Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

TEPP: A robust trust-enhanced privacy-preserving quality of service prediction method for web service recommendation

Wei-wei Wang [a,b], Wenping Ma [a,b,*], Kun Yan [a]

[a] School of Telecommunications Engineering, Xidian University, Xi'an, 710071, China
[b] State Key Laboratory of Intelligent Vehicle Safety Technology, Chongqing, 401133, China

ARTICLE INFO

Keywords:
QoS prediction
Dirichlet distribution
Privacy preservation
Evolutionary game theory
Web service recommendation

ABSTRACT

In today's service-oriented digital environment, ensuring the quality of service (QoS) is crucial, which makes QoS prediction a prominent topic in current research on Web service recommendation. Recently, some existing works have made significant advancements in modeling both users and services. However, several key issues have not been well studied in existing research, including issues related to bilateral trust, user preferences, and privacy protection. To effectively resolve these concerns, we put forward TEPP, a robust trust-enhanced privacy-preserving QoS prediction method for Web service recommendation. First, we evaluate user reputation values through the Dirichlet distribution and integrate user similarity to jointly compute trust values between users, thereby identifying a group of trustworthy and similar users. At the same time, we utilize an exponential mechanism to protect the privacy of user information. Secondly, we calculate the preference similarity between users, taking into account their preferences. Finally, we determine a set of trustworthy similar services by combining the reputation value and similarity of the service providers, and predict missing QoS by a fusion model that integrates the above three methods. To make TEPP more practical and robust in Web service recommendation, we embed a bilateral trust model in TEPP based on evolutionary game theory to constrain and guide users and service providers to honestly participate in the Web service recommendation. Experimental simulation results demonstrate that the proposed scheme not only outperforms existing schemes in prediction accuracy but also can fully motivate both users and service providers to choose trusted strategic behaviors in the Web service recommendation.

1. Inroduction

With the advancement of the Internet, significant impetus has been given to the development of areas such as cloud computing, the Internet of Things (IoT), and big data processing, establishing a solid foundation for various service-oriented downstream tasks (Chen et al., 2024; Hassan et al., 2020; Li et al., 2014; Liu & Chen, 2019; Mezni, 2023). Choosing high-quality Web services is crucial for applications like the IoT, as it not only enhances application reliability but also improves user experience, ensuring more efficient collaboration among devices (Li & Lin, 2020; Xie et al., 2019; Xu et al., 2016). Therefore, in these diverse application scenarios, research on QoS prediction becomes especially urgent to meet the pressing demands of various domains for efficient performance and exceptional user experience (Chen et al., 2020; Fiedler et al., 2010; Zheng et al., 2012b).

QoS represents the non-functional aspects of services, such as response time, availability, reliability, and throughput (Ghafouri et al., 2020; Zheng et al., 2020). These attributes can profoundly influence users' service experience and satisfaction. Hence, service providers frequently utilize QoS metrics to monitor and improve their services, ensuring they meet users' expectations and requirements. Personalized QoS prediction and Web service recommendation are crucial in today's digital environment. Predicting client QoS allows users to gain a more precise understanding of the expected performance tailored to their specific situations and needs. Moreover, Web service recommendations derived from these predicted QoS values can guide users in selecting services most likely to meet their requirements, thereby assisting them in developing high-quality applications. This approach not only enhances the user experience but also contributes to enhancing the overall efficiency and effectiveness of Web services.

Collaborative filtering is widely regarded as a fundamental technique for QoS prediction, leveraging historical behavioral patterns of users or services to estimate missing QoS values (Zheng et al., 2020). In recent years, with the continuous increase in data scale and the complexity of network services, an increasing number of studies have introduced deep learning methods to further enhance the model's capability in

* Corresponding author.
E-mail addresses: wweiwei@stu.xidian.edu.cn (W. Wang), wp_ma@mail.xidian.edu.cn (W. Ma), kyan@stu.xidian.edu.cn (K. Yan).

https://doi.org/10.1016/j.eswa.2025.128786
Received 31 December 2023; Received in revised form 16 June 2025; Accepted 24 June 2025

capturing complex interactive relationships (Zhang et al., 2024a,b; Zou et al., 2022). However, QoS data are usually uploaded by users, and their reliability is difficult to guarantee. In particular, data provided by untrusted users may introduce noise, which seriously affects prediction accuracy (Tao et al., 2012; Zheng et al., 2020). Therefore, introducing a user reputation mechanism and performing weighted evaluations of data have become important approaches to improve prediction reliability. Although existing methods have made some progress in prediction accuracy, current QoS prediction still faces the following key issues:

Bilateral trust issues. (1) The QoS values provided by certain untrustworthy users may lack reliability. (2) The existing literature hardly considers whether the service provider is trustworthy, which makes it impossible to determine whether the services it provides are beneficial to users. Hence, users are encouraged to submit feedback on service providers, enabling an assessment of the credibility of service providers through user-provided information.

User's actual preferences are ignored. User preferences are critical to QoS prediction in Web service recommendation, as they directly affect both user expectations and satisfaction. Taking these preferences into account leads to more tailored and effective service recommendations.

Privacy issues related to QoS prediction. User private information can be inferred from submitted QoS values, so how to improve prediction accuracy while ensuring user privacy remains a key challenge.

To tackle the problems mentioned above, we propose TEPP, a robust trust-enhanced privacy-preserving QoS prediction method for Web service recommendation, which is based on the Dirichlet distribution, differential privacy, and evolutionary game theory. In TEPP, the Dirichlet distribution is used to calculate the user's reputation value, and then calculate the user's trust value; differential privacy is used to protect the user's privacy; and evolutionary game theory is used to ensure the robustness of the entire system, which stabilizes the trust between users and service providers. The primary focus of this article can be summarized as follows:

- We employ the Dirichlet distribution to evaluate user reputation and incorporate an enhanced similarity measure to calculate user trust values for identifying credible similar users. Meanwhile, we introduce an exponential mechanism to protect user privacy, thereby achieving an organic integration of QoS data reliability and privacy protection.
- We propose a robust trust-enhanced privacy-preserving QoS prediction method, TEPP, which integrates QoS prediction based on trusted users, user preferences, and trusted service-oriented approach. Theoretically, we demonstrate that our fusion model adheres to ϵ-differential privacy.
- We design a bilateral trust model based on evolutionary game theory to stabilize trust relationships between users and service providers, thereby ensuring the overall robustness of the system.
- Extensive experiments on real QoS datasets verify the superiority of TEPP. Experimental results demonstrate that TEPP not only outperforms existing solutions in performance but also can effectively ensure the robustness of the model by motivating both users and service providers to choose trusted policy behaviors during the Web service recommendation.

The rest of this work is structured as follows. We briefly introduce some existing work in Section 2. Section 3 presents the basics of this paper, including the Dirichlet distribution and differential privacy. Section 4 introduces the detailed construction of the proposed scheme. Section 5 introduces the bilateral trust model. Section 6 presents comprehensive experiments along with detailed discussions and analysis. Finally, Section 7 summarizes the proposed method and new findings and shares future research directions.

2. Related work

QoS prediction, as a critical component of Web services, has gained increasing prominence due to the rapid development of cloud computing, the IoT, and 5G technology. The growing complexity and demands of networks underscore the escalating importance of QoS prediction. In this section, we conduct a concise review focusing on two main aspects, which are primarily distinguished by the presence or absence of privacy protection measures.

2.1. Traditional QoS prediction methods

QoS prediction has attracted significant attention in the field of networking research, experiencing a notable increase in associated studies and developments. Most of the earlier solutions were grounded in collaborative filtering methods (Zheng et al., 2011, 2020). Nevertheless, collaborative filtering-based prediction methods encounter challenges related to data sparsity and credibility. To address these issues, Wang et al. (2020) presented a QoS prediction scheme based on reputation-aware network embedding, which effectively mitigates the data sparsity problem and improves the prediction accuracy of collaborative filtering by constructing a user-service bipartite network. Chen et al. (2023) proposed a novel QoS prediction algorithm that transforms traditional local optimization aggregation into a global search based on swarm intelligence, enabling the prediction of missing values across the entire QoS distribution space. To achieve personalized and reliable QoS prediction in cloud environments, Liu and Chen (2019) introduced a method based on K-medoids clustering and trust-awareness for prediction and service recommendation. Su et al. (2017) presented a novel prediction method that calculates user trust values through K-means clustering and a Beta mechanism, and predicts QoS based on users or service groups with similar trust. With the development of graph representation learning, Zhu et al. (2023) applied graph contrastive learning for QoS prediction, aiming to tackle the cold start and data sparsity issues that traditional methods. Chang et al. (2021) employed the matrix factorization method of graphs to devise a QoS prediction model capable of significantly enhancing prediction accuracy. To address the limitations of traditional neural networks in QoS prediction, Zou et al. (2023) devised a flexible QoS prediction approach by fusing location-aware neural prediction techniques with neighborhood-based collaborative filtering strategies to construct an adaptive prediction model. Li et al. (2021) introduced a topology-aware neural model framework that achieves accurate QoS prediction by efficiently utilizing contextual information and avoids dependence on the underlying network topology and complex interactions of autonomous systems.

In addition, in recent years, there have been studies devoted to proposing hybrid schemes that integrate traditional collaborative filtering methods with other methods to improve the QoS prediction performance. Zheng et al. (2012a) presented a neighborhood-aware matrix factorization method that integrates similar users identified by historical QoS-based PCC into QoS prediction. He et al. (2017) proposed the NeuMF model, which combines generalized matrix factorization with multi-layer perceptron to form an integrated structure to predict missing QoS. Zhu et al. (2017) put forward a hybrid QoS prediction method that extends traditional matrix factorization by integrating user and service deviations, latent factor modeling, online learning, and adaptive weight mechanisms to improve prediction accuracy in dynamic cloud environments. Zhang et al. (2019) proposed a hybrid model LDCF that integrates the multi-layer perceptron and collaborative filtering similarity mechanism, combining location similarity modeling with the nonlinear feature extraction capability of deep learning to improve the prediction performance of complex QoS data.

2.2. Privacy-preserving QoS prediction methods

Despite advances in QoS prediction performance, growing network technologies and the popularization of privacy policies have raised awareness of personal data protection, thereby drawing scholarly attention to privacy concerns in QoS prediction.

Badsha et al. (2018a) utilized homomorphic encryption to develop a QoS prediction framework with a privacy protection function, which achieves personalized Web service recommendations without leaking privacy by encrypting historical QoS and location information. Badsha et al. (2018b) also devised a privacy-preserving Web service recommendation scheme based on homomorphic encryption, which enables QoS prediction while preserving user privacy. However, due to its high computational overhead, the scheme is primarily suitable for offline scenarios. This type of solution focuses on the performance evaluation of homomorphic encryption methods and rarely involves the research of QoS prediction accuracy. In addition to homomorphic encryption, some studies have adopted differential privacy to achieve privacy protection. Liu et al. (2019) presented a collaborative privacy protection framework for QoS prediction. They adopted the Laplace mechanism to add noise for protecting sensitive information and designed two methods: DPS, which directly adds noise to user data, and DPA, which first aggregates the data and then adds noise to improve data availability. Liu et al. (2020) introduced a shared and collaborative Web service QoS prediction scheme and developed a differential privacy framework that effectively prevents privacy leakage while enabling data sharing and collaborative prediction. To address privacy concerns in QoS prediction for mobile edge computing, Zhang et al. (2020) proposed introducing Laplace noise into the edge computing environment to achieve differential privacy and reduce the risk of user data leakage. However, this method involves a trade-off between privacy protection and prediction accuracy. Multiple calculations require injecting more noise, which in turn degrades prediction accuracy. Zhu et al. (2015) proposed a effective privacy-preserving QoS prediction framework based on data obfuscation technology and designed two classic methods: P-UIPCC and P-PMF. However, similar to Laplace-based differential privacy, this type of method suffers from the issue that increased noise reduces prediction accuracy, and it remains difficult to determine an appropriate privacy budget to balance privacy protection with prediction performance. Moreover, Perifanis and Efraimidis (2022) proposed a NeuMF architecture based on federated learning, which effectively avoids the direct exposure of original user data and can be applied to QoS prediction to achieve privacy protection. Unlike the above privacy protection schemes, our scheme adopts an exponential mechanism to maximize the accuracy of the prediction results while ensuring user privacy.

3. Preliminaries

In this section, we present the relevant basic knowledge, which includes Dirichlet distribution and differential privacy.

3.1. Dirichlet distribution

The Dirichlet distribution (Xu et al., 2019) is widely used to model multidimensional probability vectors, especially as a prior in Bayesian multinomial models. The Dirichlet distribution is commonly used to model the probability distribution of $X = \{X_1, X_2, \ldots, X_K\}$, which is a K-dimensional random variable, where K represents the number of categories or outcomes. Next, we define $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$ as a level vector, where $\theta_i \in [0,1]$ and $\theta_i < \theta_{i+1}$. Then, we use $\vec{p} = \{p_1, p_2, \ldots, p_K\}$ to express the probability distribution of X, where $P\{\theta_{i-1} < X_i < \theta_i\} = p_i$ $(1 \le i \le K)$. Suppose that $\vec{\gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$ with $\gamma_i > 0$ and $1 < i < K$. So, X is said to be a Dirichlet distribution with parameters $\vec{\gamma}$, which is denoted by $X \sim Dir(\vec{\gamma})$. Therefore, the probability density function of the Dirichlet distribution is articulated as follows:

$$f(\vec{p} \mid \vec{\gamma}) = \frac{\Gamma\left(\sum_{i=1}^{K} \gamma_i\right) \times \prod_{i=1}^{K} p_i^{\gamma_i - 1}}{\prod_{i=1}^{K} \Gamma(\gamma_i)}, \tag{1}$$

where $\Gamma(\cdot)$ denotes the Gamma function. Next, the expected value of the Dirichlet distribution can be calculated using the following formula:

$$E(p_i \mid \vec{\gamma}) = \frac{\gamma_i}{\sum_{i=1}^{K} \gamma_i}. \tag{2}$$

3.2. Differential privacy

Differential privacy (Dwork et al., 2014; Li et al., 2016) is a technology that protects individual privacy in data analysis. It introduces random noise to ensure that the addition or removal of any single data has minimal impact on the analysis results, thereby effectively preventing the leakage of sensitive information.

Definition 1 (ϵ-Differential Privacy Dwork et al., 2014)**.** A randomized algorithm $\mathcal{M}$ satisfies ϵ-differential privacy if for all neighboring datasets X and X' differing by at most one element, and for any subset of possible outputs $S \subseteq \text{Range}(\mathcal{M})$, the following inequality holds:

$$P[\mathcal{M}(X) \in S] \le exp(\epsilon) \cdot P[\mathcal{M}(X') \in S], \tag{3}$$

where ϵ is the privacy parameter that quantifies privacy protection and is referred to as the privacy budget; the smaller the value of ϵ, the stronger the privacy.

Exponential mechanism (Li et al., 2016) is a differential privacy algorithm that selects outputs through the probability distribution of a quality function to achieve a balance between privacy protection and result quality.

Definition 2 (Exponential Mechanism Li et al., 2016)**.** The exponential mechanism $\mathcal{M}_E(x, q, S)$ selects an output $r \in S$ with probability proportional to $\exp\left(\frac{\epsilon \cdot q(x,r)}{2\Delta q}\right)$, where $q(x,r)$ is a quality function and Δq is its sensitivity, which is described as follows:

$$\Delta q = \max_{r \in S} \max_{x, x' : \|x - x'\| \le 1} |q(x,r) - q(x',r)|, \tag{4}$$

where x and x' are neighboring datasets differing in at most one element, and r is a potential output.

4. Proposed method

4.1. Overview of the TEPP method

In this article, the proposed TEPP method comprises three components: a) Trusted user-based prediction method; b) User preference-based prediction method; and c) Trusted service-oriented prediction. We assert that each of the aforementioned parts can predict the values of those missing entries in the user-service matrix based on the values already present in the matrix. However, in this work, we predict the missing QoS values in the user-service matrix through an ultimate hybrid prediction method, which combines the aforementioned three methods. The overall flowchart of the proposed method is depicted in Fig. 1, and detailed descriptions of each part will be provided in the following sections. It is important to note that, for coherent exposition in Section 4, the bilateral trust model in the TEPP model will be discussed separately in Section 5.

4.2. Trusted user-based prediction

4.2.1. Reputation value calculation

We intend to use the Dirichlet distribution model to calculate the trustworthiness of users. Prior to determining a user's trust value, evaluating the satisfaction derived from invoking a specific service is essential. When assessing a user's satisfaction with a specific service, a

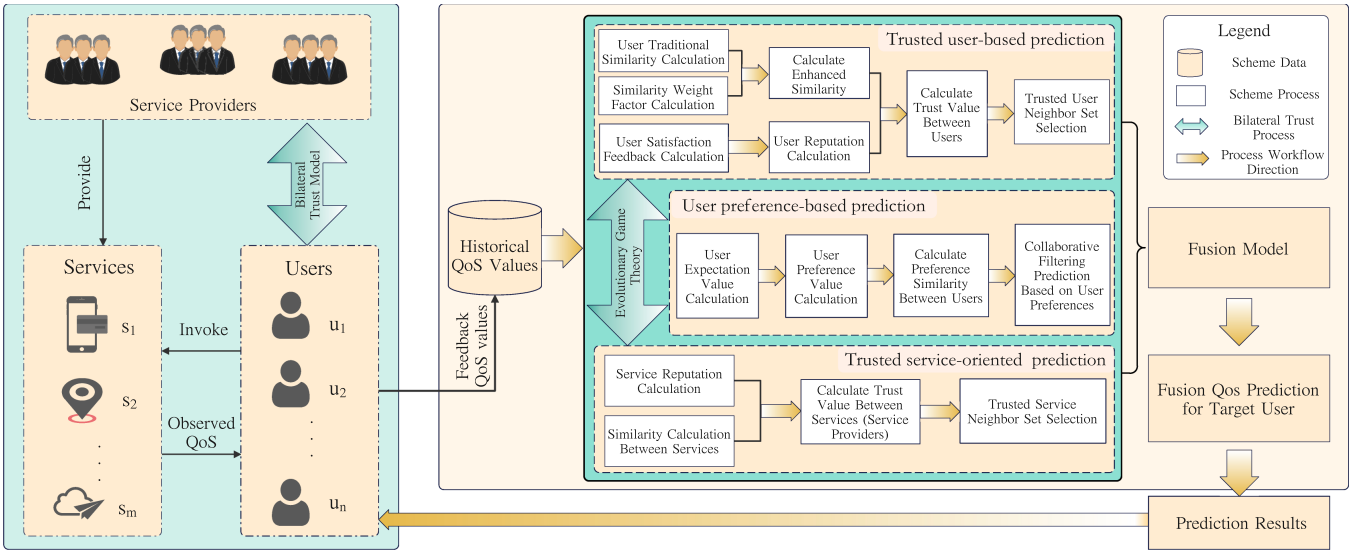

Fig. 1. Overall framework preview of TEPP. TEPP illustrates a hybrid QoS prediction framework embedded in a bilateral trust model based on evolutionary game theory. Users invoke services and provide historical QoS feedback, enabling the system to perform predictions from three aspects: trusted user-based prediction, user preference-based prediction, and trusted service-oriented QoS prediction. The upper module calculates the trust values between users to build a trusted user neighbor set for prediction; the middle module performs collaborative filtering prediction based on user expectations and preference similarity; the lower module calculates the trust values between services through service reputation and similarity to establish a trusted service neighbor set for prediction. At the same time, the bilateral trust model based on evolutionary game theory is embedded between users and service providers to motivate both parties to execute trust strategy behaviors. Finally, an adaptive fusion model generates the final QoS prediction for the target user.

weighted average can comprehensively account for various QoS attributes. Suppose there are k QoS attributes denoted as $Q_1, Q_2, \ldots, Q_k$. For each QoS attribute Q_i, collect the user's actual observation data for this attribute, denoted as $q_{i,j}$, where j denotes the jth invocation of the service. Then, a weight is assigned to each QoS attribute, indicating its importance in user satisfaction. The weight is usually a positive number and the sum is 1, expressed as $w_1, w_2, \ldots, w_k$. To standardize the observation data of each QoS attribute Q_i, we use the Z-score normalization method: $z_{i,j} = (q_{i,j} - \mu_i)/\sigma_i$, where μ_i represents the average observed value of QoS attribute Q_i: $\mu_i = \sum_{j \in I_i} q_{ij}/|I_i|$ represents QoS indicator the standard deviation of the observations of Q_i: $\sigma_i = \sqrt{\sum_{j \in I_i}(q_{ij} - \mu_i)^2/|I_i|}$, where I_i represents the set of Web services that have been invoked by user i. Based on the standardized data and weights, calculate the overall satisfaction score r: $r = \sum_{i=1}^{k} w_i \cdot z_{i,j}$, where w_i is the weight of QoS attribute Q_i, and $z_{i,j}$ is the standardized observation data. Based on the comprehensive satisfaction score r, user satisfaction can be categorized. For instance, if r is relatively high, then user satisfaction is high; otherwise, user satisfaction is low. In practical applications, to better represent user satisfaction with a particular service, we can map r to the interval $[0, 1]$. In this paper, we achieve this objective using the commonly employed Sigmoid function. The Sigmoid function is as follows:

$$S(r) = \text{Sigmoid}(r) = \frac{1}{1 + e^{-r}}. \tag{5}$$

In our TEPP method, we use the Dirichlet distribution to calculate the reputation value of a user u_i. Given a user u_i, let X be the discrete random variable defined as the satisfaction level of feedback from user u_i on the invoked service. we also denote levels of satisfaction X as a set $\{X_1, X_2, \ldots, X_K\}(X_i \in [0, 1], 1 \le i \le K, X_{i+1} > X)$. Let $\vec{p} = \{p_1, p_2, \ldots, p_K\}(\sum_{i=1}^{l} p_i = 1)$ be the probability distribution vector of X with respect to the levels of satisfaction values, and we have $P\{\theta_{i-1} < X_i < \theta_i\} = p_i (1 \le i \le K)$. Then, we also let $\vec{\gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$ denote the vector of cumulative satisfaction value. With a posterior Dirichlet distribution, $\vec{p}$ can be modeled as

$$Dir(\vec{p} \mid \vec{\gamma}) = \frac{\Gamma\left(\sum_{i=1}^{K} \gamma_i\right) \times \prod_{i=1}^{K} p_i^{\gamma_i - 1}}{\prod_{i=1}^{K} \Gamma(\gamma_i)}, \tag{6}$$

where $\gamma_0 = \sum_{i=1}^{K} \gamma_i$. To calculate the reputation value of a u_i, we assign the weight value ω_i for every level θ_i. Let p_i denote the probability that the satisfaction value of u_i's feedback is categorized into the level of θ_i, where $\vec{p} = \{p_1, p_2, \ldots, p_K\}$ and $\sum_{i=1}^{K} p_i = 1$. Let Y be the random variable denoting the weighted average of the probability of every satisfaction level in $\vec{p}$, then the reputation value $R(u_i)$ of u_i can be expressed as follows:

$$R(u_i) = E[Y] = \sum_{i=1}^{K} \omega_i E[p_i] = \sum_{i=1}^{K} \omega_i \frac{\gamma_i}{\sum_{i=1}^{K} \gamma_i} = \frac{1}{\gamma_0} \sum_{i=1}^{K} \omega_i \gamma_i. \tag{7}$$

4.2.2. Trusted neighbor selection

In this section, we aim to select a reliable set of neighbors during prediction, which will be determined by combining the computed user reputation values and the similarity between users. Building a trusted neighbor set requires calculating trust values among users. This is because the trust between users can elucidate the extent to which feedback provided by a user after invoking a particular service can be well-received and acknowledged by other users, stemming from the trust that these users have in the user providing the feedback. In our method, we quantify the consistency of the quality feedback provided by users when jointly invoking certain services by computing the similarity between them. In this article, we utilize the Pearson correlation coefficient (PCC) (Fkih, 2022; Wang et al., 2025) for computing the similarity among users. The PCC is defined as follows:

$$\text{sim}(u_i, u_j) = \frac{\sum_{a \in I_{i,j}} \left(r_{i,a} - \bar{r}_i\right)\left(r_{j,a} - \bar{r}_j\right)}{\sqrt{\sum_{a \in I_{i,j}} \left(r_{i,a} - \bar{r}_i\right)^2 \left(r_{i,a} - \bar{r}_i\right)^2}}, \tag{8}$$

where I_{ij} is the set of Web services invoked by both user u_i and user u_j, represented as $I_{ij} = I_i \cap I_j$. $r_{i,a}$ represents the QoS value of service s_a as observed by user u_i. $\bar{r}_i$ and $\bar{r}_j$ denote the average QoS values observed by users u_i and u_j for various services, respectively. To achieve higher accuracy, we have taken the help of an enhanced PCC method to introduce it into QoS prediction to calculate the trust value. The calculation method is as follows:

$$\text{En\_sim}(u_i, u_j) = \text{sim}(u_i, u_j) \times s(u_i, u_j), \tag{9}$$

where $\text{sim}(u_i, u_j)$ is the Pearson similarity between u_i and u_j, and $s(u_i, u_j)$ is a similarity weight factor. Specifically, $s(u_i, u_j)$ is defined as follows:

$$s(u_i, u_j) = T(n)^{w(u_i,u_j)}, \tag{10}$$

where $T(n) = \frac{1}{\ln(2+n)}$ with $T(n) \in (0, 1)$, n represents the count of $I_{i,j}$, and

$$w(u_i, u_j) = \sqrt{\frac{\sum_{s\in I_{i,j}} w_s (r_{is} - \bar{r}_i)^2}{\sum_{s\in I_{i,j}} w_s}}, \tag{11}$$

where $w_s = \ln(1 + \frac{t}{n_s})$, t is the total number of services, n_s denotes the total number of times service s has been invoked by all users. Subsequently, the trust value between users u_i and u_j is determined by taking into account the calculated reputation values for each user, as well as the similarity between them. The trust value is computed using the following formula:

$$\text{Trust}(u_i, u_j) = 2 \times \frac{\text{En\_sim}(u_i, u_j) \times R(u_i)}{|\text{En\_sim}(u_i, u_j)| + R(u_i)}, \tag{12}$$

where $\text{En\_sim}(u_i, u_j)$ denotes the enhanced Pearson similarity between users, and $R(u_i)$ denotes the reputation value of user u_i. And $\text{Trust}(u_i, u_j)$ denotes the trust value between user u_i and user u_j computed from the reputation value of user u_i and the improved similarity of users u_i and u_j. Indeed, it is evident from the equation that $\text{Trust}(u_i, u_j)$ ranges between -1 and 1. This is attributed to the fact that $R(u_i)$ ranges between 0 and 1, and $\text{En\_sim}(u_i, u_j)$ spans the interval $[-1,1]$. We further assert that a larger value of $\text{Trust}(u_i, u_j)$ implies that u_j is more trustworthy relative to u_i.

After computing trust values among users, the selection of the set of trusted neighbors will be carried out. Prior to this, to ensure user privacy, we will apply differential privacy in this process. To achieve this goal, we initially perform privacy neighbor set selection using the exponential mechanism. Given a set of users U, a user u_i, and a subset $\mathcal{M} \subseteq U \setminus u_i$, the quality function measuring the suitability of $\mathcal{M}$ as the neighbors of u_i is defined as:

$$q(U, u_i, \mathcal{M}) = \sum_{u_j\in\mathcal{M}} |\text{Trust}(u_i, u_j)|, \tag{13}$$

where U is the set of users in the Qos prediction system, $\text{Trust}(u_i, u_j)$ represents the reputation value between user u_i and user u_j. The quality function, as defined in Eq. (13), is the sum of the absolute values of the trust values. Based on the concept of the exponential mechanism, the probability of the set $\mathcal{M}$ being the neighbor set is obtained as follows:

$$Pr(\mathcal{M}) = \frac{exp(\frac{\epsilon}{2\Delta q} q(U, u_i, \mathcal{M}))}{\sum_{\mathcal{M}'\in U} exp(\frac{\epsilon}{2\Delta q} q(U, u_i, \mathcal{M}'))}, \tag{14}$$

where Δq represents the sensitivity of the quality function q, which is given by:

$$\Delta q = \max_{\mathcal{M}} \max_{U_1, U_2 : \|U_1 - U_2\| \le 1} \left| q(U_1, u, \mathcal{M}) - q(U_2, u, \mathcal{M}) \right| = 1, \tag{15}$$

where U_1 and U_2 represent any pair of adjacent user sets. Once the random privacy neighbor set selection is completed, we proceed to select the trusted neighbor set. Specifically, we can utilize the following equation to choose a set of reliable neighbors for user u_i:

$$N(u_i) = \{u_j | u_j \in \mathcal{M}, \text{Trust}(u_i, u_j) \ge \theta, u_i \ne u_j\}, \tag{16}$$

where $\mathcal{M}$ denotes the randomly selected privacy neighbor set obtained using differential privacy technology, and θ represents the trust threshold. Finally, the missing values in the Qos matrix are predicted by

$$R^u_{ia} = \overline{r_i} + \frac{\sum_{u_j\in N(u_i)} \text{Trust}(u_i, u_j)(r_{j,a} - \overline{r_j})}{\sum_{u_j\in N(u_i)} \text{Trust}(u_i, u_j)}, \tag{17}$$

where R^u_{ia} is the predicted QoS value of service s_a observed by user u_i. $\bar{r}_i$ represents the average QoS values observed by user u_i for various services. $\bar{r}_j$ represents the average QoS values observed by trustworthy neighbor user u_a for different services.

4.3. Trusted service-oriented prediction

In our above scheme, we utilized a user-based collaborative approach for QoS prediction. This method involved selecting a set of reliable users to constitute a trustworthy neighbor set for the target user. Then, the values within this trusted neighbor set were utilized to predict missing values in the user-services matrix for the target user. However, this approach exhibits bias by neglecting predictions from the service data perspective, resulting in slightly less accurate predictions. To enhance prediction accuracy, we propose a service-based QoS value prediction method. Simultaneously, to consider the reputation of the service provider, we indirectly represent the trustworthiness of the service provider by calculating the trust value between different services. To calculate the reputation value between different services, we first need to give the user's feedback score for the service. This is because the reputation value represents the user's inner feeling towards a certain service. It can also be viewed as explicit feedback from users who have either invoked the service in question or interacted with it in the past.

In our work, each invoked service will get a feedback rating provided by users to reflect their satisfaction level after interacting with the service. Usually, we set this feedback rating as an integer between 1 and r, and the larger the value, the higher the degree of satisfaction. Each user will give a corresponding feedback rating to express their opinion on the experience of the service. In addition, we introduce the concept of $R(s_j)$, which is used to represent the reputation of the service s_j in the whole calling process. This reputation value can comprehensively consider the performance of the service among different users, thereby forming an indication of the overall trustworthiness of the service s_j.

$$R(s_j) = \frac{\sum_{i=1}^{m} r_i}{m}, \tag{18}$$

where r_i represents the feedback score of i, and m represents the number of feedback scores.

Trust among various services reflects the high reputation values of service providers, as these services consistently receive positive feedback upon user invocation. In our work, we favor the use of similarity between different services to indicate that the explicit feedback from the user after an invocation of these services is similar. First, we employ the following formula to determine the similarity between services.

$$\text{sim}(s_a, s_b) = \frac{\sum_{u\in U_R} r_{u,a} r_{u,b}}{\sqrt{\sum_{u\in U_R} r^2_{u,a}} \sqrt{\sum_{u\in U_R} r^2_{u,b}}}, \tag{19}$$

where $U_R = \{u_i | u_i \in \mathcal{M}, R(u_i) \ge \theta\}$. $r_{i,a}$ is the QoS value of service s_a observed by user u_i. The trust value among distinct services can be computed by amalgamating the similarity among the services and the reputation value of one specific service, as outlined below:

$$T(s_a, s_b) = 2 \times \frac{R(s_a) \times \text{sim}(s_a, s_b)}{1 + |\text{sim}(s_a, s_b)|}, \tag{20}$$

where $T(s_a, s_b)$ represents the trust value between services s_a and s_b. $R(s_a)$ represents the reputation of service s_a. $\text{sim}(s_a, s_b)$ represents the similarity between services s_a and s_b. Since $R(s_a)$ belongs to [0, 1], the value range of the numerator will also be between $[-1, 1]$. Since the value range of $\text{sim}(s_a, s_b)$ belongs to $[-1, 1]$, so $|\text{sim}(s_a, s_b)|$ will also be in the range of [0, 1]. Therefore, the denominator ranges from 1 to 2. Therefore, the value range of the formula $T(s_a, s_b)$ is -1 to 1. A higher value indicates greater trustworthiness of the service providers. Upon completion of trust value calculation for all services, a set of reliable neighbors can be chosen for service s_j.

$$S(j) = \{s_t | s_t \in M(j), T(s_t, s_j) \ge \theta, s_t \ne s_j\}, \tag{21}$$

where $M(j)$ is a set of Top-S similar services to the service j. θ represents the trust threshold. $T(s_t, s_j)$ represents the trust value between services s_t and s_j. Then, the missing value in the QoS matrix based on service

prediction is calculated as follows:

$$R_{ij}^{s} = \frac{\sum_{s_b \in S(j)} r_{ib} \mathrm{Trust}(s_j, s_b)}{\sum_{s_b \in S(j)} \mathrm{Trust}(s_j, s_b)}, \tag{22}$$

where $\mathrm{Trust}(s_j, s_b)$ represents the trust value between services s_j and s_b. r_{ib} signifies the QoS value of service s_b observed by user u_i. $S(j)$ denotes the set of trustworthy neighbors of service s_j.

4.4. User preference-based prediction

QoS prediction in the trusted user-based method involves selecting a set of trustworthy neighbors for active users. However, this approach fails to consider the actual preferences of users. Next, in this part, we propose a method that relies on user preferences to predict QoS values. The following details outline our approach: Initially, we calculate users' preferences for different services based on various QoS attributes. This process includes taking into account the observed values of the service on QoS attributes and the user's expected values, allowing us to derive the user's preference value. The specific expression is as follows:

$$p_{i,j} = \frac{(1-\eta) r_{i,j} + \eta r_i e_{i,j}}{r_i}, \tag{23}$$

where $p_{i,j}$ represents user u_i's preference for service j on QoS attribute k. r_i denotes the average QoS values observed by users u_i for various services. $r_{i,j}$ represents the QoS value of service s_j as observed by user u_i. For the expected value $e_{i,j}$, we calculate it using the following method. A group of the Top-m most trustworthy users is identified as the neighbor set $N(u_i)$ of user u.

$$e_{i,j} = \frac{\sum_{u \in N(u_i)} r_{i,j} \mathrm{sim}(u_i, u)}{\sum_{u \in N(u_i)} \mathrm{sim}(u_i, u)}, \tag{24}$$

where $e_{i,j}$ is user u_i's expectation value for service j on a certain QoS attribute. $N(u_i)$ represents the most trustworthy neighbor set of user u_i. $\mathrm{sim}(u_i, u)$ is the Pearson similarity between u_i and u. Upon acquiring the user's preference values, we compute the similarity in preferences between users, a step crucial for predicting missing values later on. The calculation of preference similarity is based on the preference values. Therefore, we opt for Euclidean distance to calculate preference similarity between users u_i and u_j, as illustrated below:

$$\mathrm{sim}_{pre}(u_i, u_j) = \frac{1}{1 + \sqrt{\sum_{s \in S(j)} (p_{u_i,s} - p_{u_j,s})^2}}, \tag{25}$$

where $\mathrm{sim}_{pre}(u_i, u_j)$ represents the preference similarity between user u_i and user u_j, $p_{u_i,s}$ represents user u_i the preference value of service s on a specific QoS attribute, $p_{u_j,s}$ represents the preference value of user u_j on a specific QoS attribute, $S(j)$ represents the trustworthy neighbor set for service s_j by Eq. (21). Next, the missing QoS value prediction based on user preferences is performed. The specific calculation method is as follows:

$$R_{ij}^{p} = \overline{r_i} + \frac{\sum_{u_a \in N(u_i)} \mathrm{sim}_{pre}(u_i, u_a)(r_{aj} - \overline{r_a})}{\sum_{u_a \in N(u_i)} \mathrm{sim}_{pre}(u_i, u_a)}, \tag{26}$$

where $N(u_i)$ represents the set of user neighbors similar to user u_i. By following the aforementioned steps, a method grounded in user preferences can be utilized to calculate the QoS prediction value for a particular service. This will help make personalized service recommendations and predict the user's missing values for services based on the user's preference similarity.

4.5. Fusion QoS prediction for target users

In the initial part of this section, we pointed out that while the three methods introduced earlier can all conduct QoS prediction, none of them can attain a high level of prediction accuracy. Consequently, to achieve optimal prediction performance for missing QoS values, we intend to employ a fusion model that unifies these three complementary methods. This constitutes the proposed TEPP prediction model, which fully leverages the strengths of all three methods. Building upon the aforementioned three fusion prediction methods, the proposed TEPP method employs two parameters μ and ν, to integrate these methods into a linear fusion prediction model. The specific form of the fusion model is shown in Eq. (27), which is designed to achieve superior prediction accuracy.

$$R_{ij} = \mu R_{ij}^{u} + \nu R_{ij}^{s} + (1 - \mu - \nu) R_{ij}^{p}, \tag{27}$$

where R_{ij} denotes the final predicted QoS value. The parameters μ and ν indicate the relative contributions of the TEPP method to R_{ij}^{u} and R_{ij}^{s}, respectively, and the value range of the two parameters is $0 \le \mu + \nu \le 1$.

4.6. Privacy analysis

In this part, we will theoretically prove that our proposed method satisfies ϵ-differential privacy in Theorem 1.

Theorem 1. *The proposed TEPP satisfies ϵ-differential privacy.*

Proof. We iteratively construct candidate neighbor sets of size M until all such sets within the target set have been generated. By aggregating these candidate sets, we obtain the final neighbor set $\mathbb{M}$. For any two neighboring datasets U_1 and U_2 and any $M \in \mathbb{M}$, we have

$$\frac{\exp\left(\frac{\varepsilon q(U_1,u,M)}{2\Delta q}\right)}{\exp\left(\frac{\varepsilon q(U_2,u,M)}{2\Delta q}\right)} = \exp\left(\frac{\epsilon\big(q(U_1,u,M) - q(U_2,u,M)\big)}{2\Delta q}\right) \le \exp(\frac{\epsilon}{2}).$$

Analogously, the following inequality also holds:

$$\exp\left(\frac{\varepsilon q\big(U_2,u,M'\big)}{2\Delta q}\right) \le \exp(\frac{\epsilon}{2}) \cdot \exp\left(\frac{\varepsilon q\big(U_1,u,M'\big)}{2\Delta q}\right).$$

Based on the definition of the exponential mechanism, we evaluate the probability ratio that output $M \in \mathbb{M}$ occurs on neighboring datasets U_1 and U_2. Specifically, we have:

$$\Pr[\mathcal{M}_q^{\epsilon}(U_1) = M] = \frac{\exp(\frac{\varepsilon q(U_1,u,M)}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}$$

and

$$\Pr[\mathcal{M}_q^{\epsilon}(U_2) = M] = \frac{\exp(\frac{\varepsilon q(U_2,u,M)}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_2,u,M')}{2\Delta q})}.$$

Hence, the probability ratio is given by:

$$\begin{aligned}
&\frac{\Pr[\mathcal{M}_q^{\epsilon}(U_1) = M]}{\Pr[\mathcal{M}_q^{\epsilon}(U_2) = M]} \\
&= \left(\frac{\exp(\frac{\varepsilon q(U_1,u,M)}{2\Delta q})}{\exp(\frac{\varepsilon q(U_1,u,M)}{2\Delta q})}\right) \cdot \left(\frac{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}\right) \\
&= \exp\left(\frac{\epsilon(q(U_1,u,M) - q(U_2,u,M))}{2\Delta q}\right) \cdot \left(\frac{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_2,u,M')}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}\right) \\
&\le \exp(\frac{\epsilon}{2}) \cdot \left(\frac{\sum_{M' \in \mathbb{M}} \exp(\frac{\epsilon}{2}) \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}\right) \\
&\le \exp(\frac{\epsilon}{2}) \cdot \exp(\frac{\epsilon}{2}) \cdot \left(\frac{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}{\sum_{M' \in \mathbb{M}} \exp(\frac{\varepsilon q(U_1,u,M')}{2\Delta q})}\right) \\
&= \exp(\epsilon).
\end{aligned}$$

Therefore, we demonstrate that the TEPP scheme adheres to ϵ-differential privacy. It is noteworthy that within the three components

of the fusion model, the exponential mechanism is exclusively employed during the trusted neighbor selection phase of the trust-based user QoS prediction method. Additionally, the latter two methods solely rely on the output outcomes of the initial one. Consequently, considering the post-processing features inherent in differential privacy, TEPP indeed meets the criteria for differential privacy.

5. Bilateral trust model based on evolutionary game theory

The bilateral trust model primarily aims to tackle two aspects of trust in Web service recommendation systems: user trust in the platform or service provider and the service provider's trust in the user. This section concentrates on exploring bilateral trust issues using evolutionary game theory. In general, trust-based evolutionary game theory models encourage both parties to continuously adjust their strategies by exploring each other's trust values in the service recommendation. Therefore, by establishing a bilateral trust model, it is possible to analyze and constrain the trust behaviors of both parties during the prediction process of recommendation services.

(1) **Model and payoff matrix:** In this part, we define the bilateral trust evolutionary game model based on reputation as a 4-tuple $\mathcal{G} = \{P, G, T, N\}$, where

- P: The participants in the trust game (including users and service providers) under the bilateral trust model in the web service recommendation process can dynamically decide whether to engage in trust behavior. Assume that the two parties participating in the game are user U and service provider S, respectively.
- G: A collection of relevant trust behavior participants.
- T: The behavioral strategy space from which both parties in the game can select $T = \{t_1, t_2\} = \{\text{Trust}, \text{Mistrust}\}$.
- N: The payoff matrix is obtained from the trust game between the participating parties, as illustrated in Table 1.

It is obvious from the benefit matrix that the strategies that both parties can choose under this model are trust or mistrust. Therefore, the game process between the two parties essentially follows a "two-party symmetric game." To analyze the evolutionary game process in detail, U_i and S_i represent the two parties in the game; B_i is employed to calculate the benefits of U or S respectively. The specific calculation is as follows:

$$B_i = \zeta_1 \ln(1 + |T_i|) + \zeta_2 M_i \tag{28}$$

where $\zeta_1 + \zeta_2 = 1$, M_i represents user or service provider satisfaction, signifying either the user's satisfaction with the service or the service provider's satisfaction with the user. T_i can be calculated based on T_u and T_s, which can be calculated based on Eqs. (12) and (20), respectively.

In the Web service recommendation process, the willingness of users or service providers to engage in trusting behaviors is influenced by various factors, including user preferences, time, and the associated costs. For this purpose, we introduce a dynamic coefficient denoted as κ_i. If fewer participants engage in trust behaviors, we set κ_i greater than 1 to encourage active participation in the trust game by users or service providers; otherwise, we can set $0 < \kappa < 1$ to prevent malicious participants from destroying the entire game system.

In addition, the goal of this model is to encourage more users and service providers to honestly participate in the recommendation service process. When both parties initially participate, even if they have a good reputation but are forced to be unprofitable, in situations like this the participants may be inclined not to participate or to linger. Therefore, to encourage more users or service providers to participate, the model will be set up so that when there are fewer participants, part of the rewards will be given to those participating in the bilateral trust. This dynamic income from rewards is represented by A. Participants in the bilateral trust model need to pay a certain price to gain the trust of the other party. These costs include the time cost of calculating your own reputation value information and the cost of leaking your own private information, etc. We record this cost consumed in the process of participating in the game as C. Naturally, if one chooses to behave in a distrustful manner, there is neither a cost nor a corresponding reward.

Table 1
Payoff matrix.

The Strategy of the User	The Strategy of the Service Provider	
	Trust	Mistrust
Trust	$\kappa_u B_u + A_u - C_u$, $\kappa_s B_s + A_s - C_s$	$B_u + A_u - C_u$, 0
Mistrust	0, $B_s + A_s - C_s$	0, 0

(2) **Dynamic equation:** Assuming that all behavioral participants in this model system constitute a group G, and each participant's probability of choosing a trusting behavior strategy is denoted as p, where p is a function concerning time t and $p \in [0, 1]$. In contrast, the probability of a participant choosing a mistrusting behavior is $1 - p$. Utilizing the payoff matrix presented in Table 1 and guided by the principles of evolutionary game theory (Phelps & Wooldridge, 2013), one can derive the expected benefits of whether the user or service provider chooses trust behavior can be obtained by the following formulas:

$$E(t_1, p) = p(\kappa_i B_i + A_i - C_i) + (1 - p)(B_i + A_i - C_i), \tag{29}$$

and

$$E(t_2, p) = 0, \tag{30}$$

respectively. Therefore, the average revenue of users or service providers can be calculated as follows:

$$\begin{aligned} \overline{E} &= pE(t_1, p) + (1 - p)E(t_2, p) \\ &= p[p(\kappa_i B_i + A_i - C_i) + (1 - p)(B_i + A_i - C_i)]. \end{aligned} \tag{31}$$

Hence, the growth rate of the trust strategy can be expressed by the replication dynamic equation of the participants in the trust game, as illustrated below:

$$F_i(p) = \frac{dp}{dt} = p[E(t_1, p) - \overline{E}]. \tag{32}$$

Let $F_i(p) = 0$. Then, we can get three steady-state points that the replication dynamic equation, which are $p_1 = 0$, $p_2 = 1$, and $p_3 = \frac{C_i - B_i - A_i}{(\kappa_i - 1)B_i}$, respectively.

(3) **Dynamic and steady state analysis:** The evolutionary stable strategy (ESS) theory emphasizes that the stable state of the dynamic system should persist stable in the face of minor disturbances. In short, if a certain point p is defined as an evolutionary equilibrium point (EEP), then the derivative of the corresponding evolutionary strategy $F_i(p)$ must satisfy $F_i^{'}(p) < 0$. Based on this theory, we can infer the evolutionary strategy (ES) selected by the participants by analyzing the three stable points in different situations. See Table 2 for details.

Then, we can analyze the steady state strategies and dynamic strategies included in the trust game through Table 2. The specific classification statistics are in Table 3. However, factors that influence evolutionary trends cover participation costs, dynamic coefficient, and the initial proportion of users or service providers performing trusting behaviors in QoS predictions. With other parameters held constant, the dynamic coefficient is a key factor in increasing the proportion of participants taking trusting actions. According to Table 2, the steady state result remains constant in the game outcomes, unaffected by parameter changes. Therefore, We mainly analyze the following five dynamic strategies: II-(2), III-(2), VI, VIII-(1), and IX-(1). In dynamic strategies II-(2) and III-(2), since the EEP is not a certain value, we set $p_3 = \frac{C_i - B_i - A_i}{(\kappa_i - 1)B_i}$ tends to 1, that is, when κ_i tends to $\frac{C_i - A_i}{B_i}$, users or service providers are more willing to perform trusting behaviors.

However, for the other three remaining dynamic strategies, we can get the value range of the dynamic coefficient κ_i for participants to perform trusting behaviors as follows:

$$\frac{C_i - B_i - A_i}{(\kappa_i - 1)B_i} < p^*, \tag{33}$$

Table 2
Analysis and classification of evolutionary stable strategies.

Range of κ_i	Strategy No.		Range of parameters		Evolutionary strategy analysis
$0<\kappa_i<1$	Strategy I		$B_i + A_i > B_i > \kappa_i B_i > C_i$		$F'(p_1) > 0$, $F'(p_2) < 0$, $p_3 > 1$ does not exit EEP: p_2; ES: engaging the action of trust
	Strategy II	(1)	$B_i + A_i > B_i > C_i > \kappa_i B_i$	$\kappa_i B_i + A_i > C_i$	$F'(p_1) > 0$, $F'(p_2) < 0$, $F'(p_3) > 0$ EEP: p_2; ES: engaging the action of trust
		(2)		$\kappa_i B_i + A_i < C_i$	$F'(p_1) > 0$, $F'(p_2) > 0$, $F'(p_3) < 0$ EEP: p_3; ES: p_3 the ratio of players choosing trust action
	Strategy III	(1)	$B_i + A_i > C_i > B_i > \kappa_i B_i$	$\kappa_i B_i + A_i > C_i$	$F'(p_1) > 0$, $F'(p_2) < 0$, $F'(p_3) > 0$ EEP: p_2; ES: engaging the action of trust
		(2)		$\kappa_i B_i + A_i < C_i$	$F'(p_1) > 0$, $F'(p_2) > 0$, $F'(p_3) < 0$ EEP: p_3; ES: p_3 the ratio of players choosing trust action
	Strategy IV		$C_i > B_i + A_i > B_i > \kappa_i B_i$		$F'(p_1) < 0$, $F'(p_2) > 0$, $p_3 < 0$ does not exit EEP: p_1; ES: not engaging the action of trust
$\kappa_i > 1$	Strategy V	(1)	$\kappa_i B_i > B_i + A_i > B_i > C_i$		$F'(p_1) > 0$, $F'(p_2) < 0$, $p_3 < 0$ does not exit EEP: p_2; ES: engaging the action of trust
		(2)	$B_i + A_i > \kappa_i B_i > B_i > C_i$		
		(3)	$B_i + A_i > \kappa_i B_i > C_i > B_i$		
		(4)	$\kappa_i B_i > B_i + A_i > C_i > B_i$		
	Strategy VI		$\kappa_i B_i > C_i > B_i + A_i > B_i$		$F'(p_1) < 0$, $F'(p_2) < 0$, $F'(p_3) > 0$ EEP: p_1,p_2; ES: $p_i \in (0, p_3)$ not engaging the action of trust $p_i \in (p_3, 1)$ engaging the action of trust
	Strategy VII	(1)	$B_i + A_i > C_i > \kappa_i B_i > B_i$	$\kappa_i B_i + A_i > C_i$	$F'(p_1) > 0$, $F'(p_2) < 0$, $p_3 < 0$ does not exit EEP: p_2; ES: engaging the action of trust
		(2)		$\kappa_i B_i + A_i < C_i$	$F'(p_1) > 0$, $F'(p_2) > 0$, $F'(p_3) > 0$ EEP: No;
	Strategy VIII	(1)	$C_i > B_i + A_i > \kappa_i B_i > B_i$	$\kappa_i B_i + A_i > C_i$	$F'(p_1) < 0$, $F'(p_2) < 0$, $F'(p_3) > 0$ EEP: p_1,p_2; ES: $p_i \in (0, p_3)$ not engaging the action of trust $p_i \in (p_3, 1)$ engaging the action of trust
		(2)		$\kappa_i B_i + A_i < C_i$	$F'(p_1) < 0$, $F'(p_2) > 0$, $p_3 > 1$ does not exit EEP: p_1; ES: not engaging the action of trust
	Strategy IX	(1)	$C_i > \kappa_i B_i > B_i + A_i > B_i$	$\kappa_i B_i + A_i > C_i$	$F'(p_1) < 0$, $F'(p_2) < 0$, $F'(p_3) > 0$ EEP: p_1,p_2; ES: $p_i \in (0, p_3)$ not engaging the action of trust $p_i \in (p_3, 1)$ engaging the action of trust
		(2)		$\kappa_i B_i + A_i < C_i$	$F'(p_1) < 0$, $F'(p_2) > 0$, $p_3 > 1$ does not exit EEP: p_1; ES: not engaging the action of trust

Table 3
Evolutionary state classification from Table 2.

State	Strategies
Steady	I, II-(1), III-(1), IV, V, VII-(1), VIII-(2), IX-(2)
Dynamic	II-(2), III-(2), VI, VIII-(1), IX-(1)

where p^* represents the initial proportion of participants who perform trusting behavior. By transforming the above formula, the value range of κ_i can be obtained: $\kappa_i > \frac{C_i - B_i - A_i}{p^* B_i} + 1$. It can be seen that in dynamic strategies VI, VIII-(1), and IX-(1), we can use the lower limit of κ_i to ensure that participants perform trusting behaviors under smaller rewards. To be more specific, we set κ_i as follows:

$$\begin{aligned}\kappa_i &= \frac{C_i - B_i - A_i}{p^* B_i} + 2 \\ &= \frac{C_i - [\zeta_1 ln(1 + |T_i|) + \zeta_2 M_i] - A_i}{p^*[\zeta_1 ln(1 + |T_i|) + \zeta_2 M_i]} + 2.\end{aligned} \tag{34}$$

Therefore, the relationship between κ_i, the trust value of the participants, M_i, and the initial proportion of participants engaging in trusting behaviors is dynamic. At the same time, given the above analysis, to encourage users and service providers to perform trusting behaviors in the trust game, the following strategy can be adopted: when the proportion of participants who initially perform trusting behaviors is lower than p_3, the system can continuously increase the reward income A_i to increase the participant ratio. Here, the interval lower bound of A_i can be obtained by transforming Eq. (33): $A_i > C_i - [(\kappa_i - 1)p^* + 1]B_i$, to fit the actual situation, we always hope that the reward with the least expenditure will urge participants to actively perform trusting behaviors in the trust game model, so we set the reward value to the lowest effective value of the interval: $A_i = C_i - [(\kappa_i - 1)p^* + 1][\zeta_1 \ln(1 + |T_i|) + \zeta_2 M_i] + 1$. It is essential to note that the values provided strive to closely reflect the actual situation, and the extent of increase is influenced by numerous factors. Moreover, if the participant ratio surpasses p_3, there is no necessity to introduce extra rewards at this time. At this point, the system can reach the maximum participation ratio.

6. Experiment

6.1. Data description and experimental setup

The dataset used in our experiment comes from the public WS-DREAM[1] database published by Zheng and Lyu (2010), Zheng et al. (2011), which is the most widely used standard dataset in the field of QoS prediction research and has excellent representativeness and versatility. We selected two datasets from the WS-DREAM database, namely Dataset#1 (D1) and Dataset#2 (D2), for model performance evaluation and comparative experiments.

To verify the performance of our proposed TEPP model in QoS prediction tasks, we first conducted a systematic experiment on the widely

[1] https://inpluslab.com/wsdream/

used D2 dataset. This dataset contains two QoS attributes: response time (RT) and throughput (TP), covering invocation records from 339 users to 5825 Web services, totaling 1,974,675 historical QoS data. The D2 dataset exhibits high sparsity and better simulates the complex service invocation environments encountered in real-world scenarios. It has been widely used in existing studies and serves as a representative dataset for validating QoS prediction schemes.

To further evaluate the generalization and scalability of the TEPP model under different data scales and distribution conditions, we conducted experiments on datasets D1 and a subset of D2 (referred to as D2-Sub-RT). D1 contains 150 user invocation records for 100 Web services, a total of 150 files, each file corresponds to the test results of a user node, and records 10,000 invocations of the node to all 100 services. Each record includes attributes such as Client IP, service ID, RT, etc., with a total of more than 1.5 million data records. Due to the large RT values in the D1 dataset, we utilized the z-score standardization method for normalization. D2-Sub-RT is a sequentially selected subset of the RT dataset from D2, containing 120 users and 5000 services, covering a total of 600,000 historical QoS data, to simulate a real-world scenario that is much more sparse and where the size of the service is significantly larger than the number of users.

In addition, considering the highly sparse nature of user-service invocation data in practical applications, this paper systematically evaluates the model's QoS prediction performance across five different extremely low data densities (0.1 %, 0.2 %, 0.3 %, 0.4 %, and 0.5 %). Under each density setting, the training set consists of a corresponding proportion of QoS samples, and the remaining data is used as a test set to fairly compare the performance of various methods in a sparse data environment. The specific experimental parameter settings are as follows: $\theta = 0.5$, $m = 30$, $S = 50$, $\epsilon = 1$ for TEPP.

All comparative experiments in this study were conducted on a standalone machine running Python 3.8 within the PyCharm IDE. The system was powered by an Intel Core i5-11400F CPU at 2.60 GHz, equipped with 32 GB of memory, and an NVIDIA GeForce RTX 3080 Ti GPU.

6.2. Evaluation metrics

To evaluate the predictive performance of TEPP against other approaches, three common accuracy metrics are employed to assess the deviation between the actual and predicted values on the test set. The two evaluation metrics formulas are described as follows: mean absolute error (MAE) and root mean square error (RMSE). MAE is defined as

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |q_{u,s} - \overline{q}_{u,s}|}{N}, \tag{35}$$

RMSE is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (q_{u,s} - \overline{q}_{u,s})^2}{N}}, \tag{36}$$

where $q_{u,s}$ and $\overline{q}_{u,s}$ represent true value and the predicted value of user u invoking service s, respectively. N denotes the number of predicted QoS values.

6.3. Compared methods

To assess the effectiveness of TEPP, we carried out a series of comparative experiments involving fourteen diverse competing approaches, which are presented below. In general, these methods cover traditional collaborative filtering approaches such as UMEAN (Sarwar et al., 2001), UPCC (Shao et al., 2007), UIPCC (Zheng et al., 2011), and NRCF (Sun et al., 2012); matrix factorization techniques including PMF (Salakhutdinov & Mnih, 2007), NIMF (Zheng et al., 2012a), and AMF (Zhu et al., 2017); deep learning-based models such as NeuMF (He et al., 2017), LDCF (Zhang et al., 2019), and BGCL (Zhu et al., 2023); as well as a variety of privacy-preserving methods including P-UIPCC (Zhu et al., 2015), P-PMF (Zhu et al., 2015), Lap-UCF (Zhang et al., 2020), and Fed-NeuMF (Perifanis & Efraimidis, 2022); Some hybrid approaches, such as UIPCC, AMF, NeuMF, and LDCF, are also incorporated, thereby providing a comprehensive and balanced basis for comparison.

- **UMEAN** (Sarwar et al., 2001): This scheme predicts empty QoS values by averaging the existing QoS of the target user.
- **UPCC** (Shao et al., 2007): This method identifies similar users using PCC and predicts missing QoS values based on their past service interactions information.
- **UIPCC** (Zheng et al., 2011): This method is a hybrid collaborative filtering scheme that combines users and items, and introduces confidence weight linear fusion to achieve QoS prediction for target users.
- **NIMF** (Zheng et al., 2012a): This approach integrates neighborhood-based similarity information into the matrix factorization framework, aiming to improve QoS prediction accuracy by jointly modeling global latent factors and local neighbor influences.
- **NeuMF** (He et al., 2017): This approach entails an advanced neural collaborative filtering technique that integrates multi-layer perceptrons with generalized matrix factorization within recommender systems.
- **NRCF** (Sun et al., 2012): This approach focuses on personalized web service recommendation by leveraging a refined collaborative filtering framework, where a new similarity measure is proposed to improve the precision of user and service matching.
- **PMF** (Salakhutdinov & Mnih, 2007): This method mainly uses probabilistic matrix factorization to optimize the traditional matrix factorization scheme to predict QoS values.
- **AMF** (Zhu et al., 2017): This approach is a hybrid QoS prediction method that integrates auxiliary information matrix and traditional matrix factorization technology, aiming to improve the prediction accuracy under sparse data conditions.
- **LDCF** (Zhang et al., 2019): This method combines the location similarity modeling in traditional collaborative filtering with the advantages of deep learning in capturing nonlinear features for QoS prediction.
- **P-PMF** (Zhu et al., 2015): This method embeds the differential privacy mechanism into the probabilistic matrix factorization (PMF) model and achieves privacy protection by perturbing the QoS data uploaded by users.
- **P-UIPCC** (Zhu et al., 2015): This method adds noise disturbance to the original QoS data based on the traditional weighted user-item collaborative filtering method to achieve QoS prediction while protecting user privacy.
- **Lap-UCF** (Zhang et al., 2020): This method combines the Laplace mechanism with the idea of user-based collaborative filtering, aiming to achieve QoS prediction in mobile edge environments while effectively protecting data privacy.
- **FedNeuMF** (Perifanis & Efraimidis, 2022): This approach is a hybrid QoS prediction method that integrates neural matrix factorization with federated learning, balancing nonlinear feature modeling and user data privacy.
- **BGCL** (Zhu et al., 2023): This approach is a QoS prediction framework for Web services using graph contrastive learning.

6.4. Sensitivity analysis of the parameters

In our fusion model, we define two parameters μ and ν. The parameters μ and ν reflect the relative importance of our proposed TEPP method to R_{ij}^{u} and R_{ij}^{s}, respectively. In this section, to investigate the impact of these two parameters on the prediction performance of our scheme, we conducted experiments with the parameters varying from 0 to 1 on the D2 dataset. The results are depicted in Fig. 2, which illustrates the comparative performance of RT and TP under different values of parameters μ and ν. As can be seen in Fig. 2, user preferences play a role in achieving performance. Specifically, for RT, the optimal settings of parameters μ

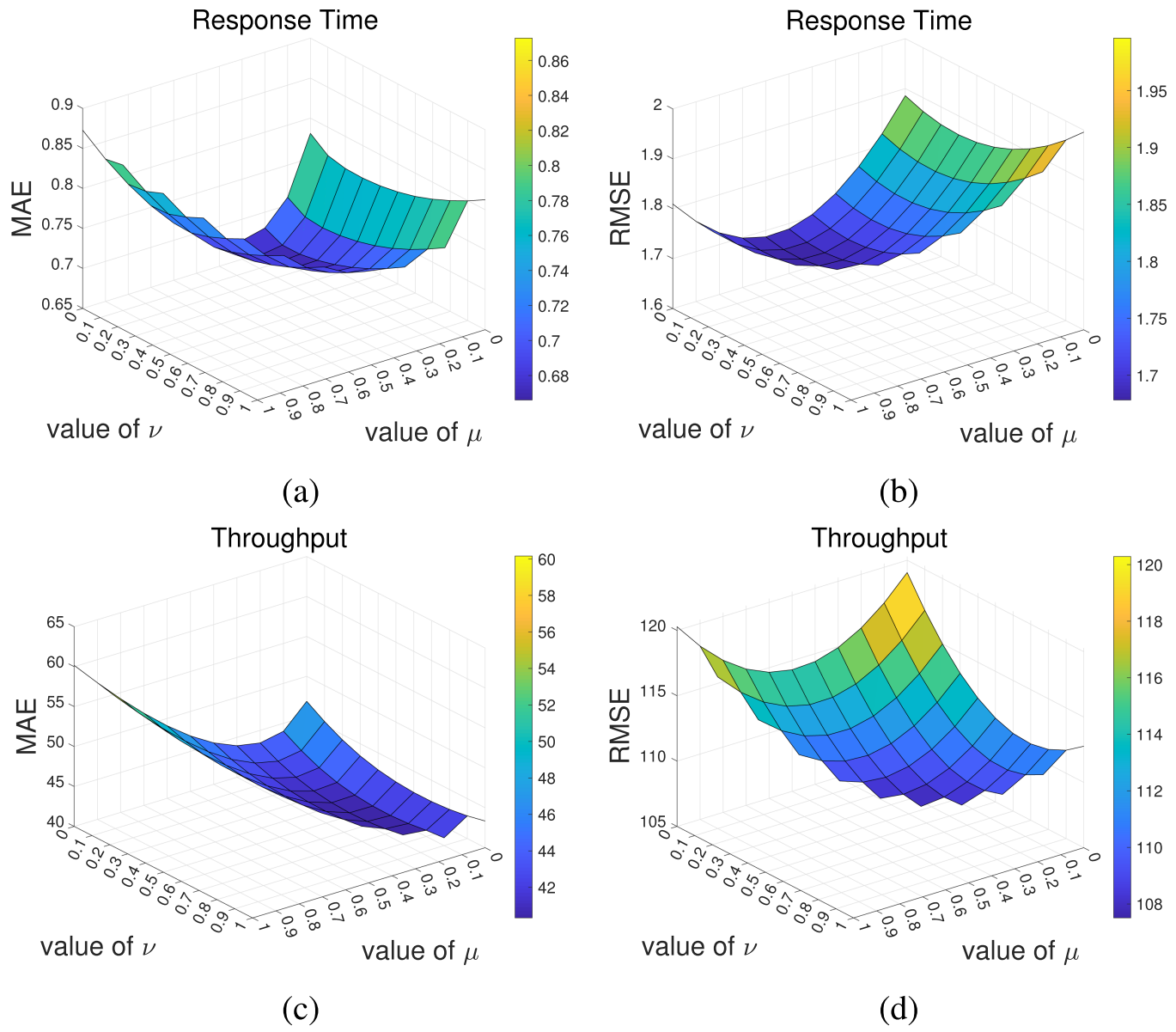

Fig. 2. Parameter sensitivity analysis of μ and ν of TEPP under different data types.

and ν are $\mu = 0.3$ and $\nu = 0.1$, and for TP, the optimal settings are $\mu = 0.1$ and $\nu = 0.6$. Meanwhile, it can be observed from the figure that the values at the image edges are larger, which occurs when $\mu = 0$, $\nu = 0$, or $\mu + \nu = 1$. This scenario indicates that the impact of one of the solutions is lost in the fusion model, leading to poor prediction performance. This also indicates that the fusion model achieves the best prediction performance, and the three schemes are interdependent. Therefore, in the subsequent comparison experiments and ablation experiments, we set $\mu = 0.3$ and $\nu = 0.1$ for RT, and $\mu = 0.1$ and $\nu = 0.6$ for TP.

6.5. Experimental result and analysis

To assess the performance of our proposed TEPP, we conduct two parts of experiments, including comparisons with state-of-the-art solutions on three datasets and ablation experiments on the D2 dataset. The specific experiments and analysis are discussed as follows.

6.5.1. Comparative experiments and analysis

In the comparative experiments, historical QoS records were divided into five different densities, including 0.1 %, 0.2 %, 0.3 %, 0.4 %, and 0.5 %, respectively. All comparative experiments were evaluated on the same dataset or a consistent subset to ensure the consistency of the comparative experiments, and the performance of QoS prediction was evaluated on the test set using MAE and RMSE metrics. To maintain fairness in evaluating prediction performance, we executed TEPP multiple times and reported the average results.

To validate the prediction performance of TEPP, we first conducted comparative experiments with advanced solutions on the widely used D2 dataset. Tables 4 and 5 show the MAE and RMSE results of each solution for response time (RT) and throughput (TP) prediction on this dataset, respectively. In the QoS prediction task, the smaller the values of MAE and RMSE, the better the prediction performance of the model. It can be clearly seen from the experimental results that TEPP is significantly better than the existing competing methods in both RT and TP. UMEAN performs poorly because it only uses historical averages as prediction results and lacks modeling of user behavior or service characteristics. UPCC and UIPCC use collaborative filtering methods, based on user neighborhoods or user-service dual neighborhoods, respectively, to significantly improve prediction accuracy. PMF, as a basic matrix factorization method, has significantly improved prediction performance

Table 4
Performance comparison of different QoS prediction methods for Response Time on the D2 dataset.

Method	Density = 0.1 %		Density = 0.2 %		Density = 0.3 %		Density = 0.4 %		Density = 0.5 %	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.9857	2.0803	0.9389	1.9355	0.9072	1.9065	0.8968	1.8899	0.8961	1.8890
UPCC	0.9663	2.0585	0.9480	1.9422	0.9038	1.9049	0.9035	1.8925	0.8853	1.8896
UIPCC	0.9081	2.1952	0.8942	2.1330	0.8841	2.0630	0.8883	1.9754	0.8850	1.8868
AMF	0.9555	1.9678	0.7999	1.8228	0.7336	1.7804	0.6994	1.7601	0.6798	1.7545
PMF	0.9044	2.1392	0.8932	2.1150	0.8887	2.1053	0.8813	2.0880	0.8777	2.0826
NRCF	0.8884	**1.8333**	0.8448	**1.7287**	0.8342	**1.7139**	0.8291	**1.7046**	0.8266	**1.7023**
NIMF	0.8809	2.0826	0.8754	2.0765	0.8564	2.0417	0.8807	2.0868	0.8775	2.0635
NeuMF	**0.7295**	2.0097	0.7139	1.9954	0.6991	1.9778	0.6961	1.9622	0.6876	1.9577
LDCF	0.8500	2.1003	0.8392	2.3665	0.7595	2.0402	0.7436	1.9724	0.7283	1.9158
BGCL	0.7386	1.8429	**0.7016**	1.7783	**0.6864**	1.7506	**0.6664**	1.7355	**0.6481**	1.7071
TEPP	**0.6660**	**1.7211**	**0.6426**	**1.6855**	**0.6307**	**1.6808**	**0.6273**	**1.6799**	**0.6126**	**1.6665**
Gains	8.70 %	6.12 %	8.41 %	2.50 %	8.11 %	1.93 %	5.87 %	1.45 %	5.48 %	2.10 %
P-UIPCC	0.9685	2.0052	0.9191	1.8697	0.9016	1.8400	0.8939	1.8362	0.8887	1.8372
P-PMF	0.9420	1.9676	0.9060	1.8486	0.8967	1.8298	0.8716	1.8248	0.8647	1.8192
Lap-UCF	0.9853	**1.9487**	0.9108	**1.8385**	0.8824	**1.7975**	0.8576	**1.7792**	0.8439	**1.7699**
FedNeuMF	**0.7391**	2.0506	**0.7241**	2.0256	**0.7156**	2.0112	**0.7048**	1.9893	**0.6976**	1.9889
TEPP-ϵ	**0.6709**	**1.7273**	**0.6522**	**1.7245**	**0.6411**	**1.7014**	**0.6319**	**1.6995**	**0.6236**	**1.6845**
Gains	3.41 %	11.07 %	3.60 %	5.70 %	3.73 %	4.81 %	3.65 %	3.98 %	3.70 %	4.27 %

Table 5
Performance comparison of different QoS prediction methods for Throughput on the D2 dataset.

Method	Density = 0.1 %		Density = 0.2 %		Density = 0.3 %		Density = 0.4 %		Density = 0.5 %	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	60.6658	121.6744	58.3526	115.8634	55.3294	112.8759	55.8713	112.9005	56.5893	112.7764
UPCC	59.9779	119.8487	55.0104	114.8311	57.2795	113.8211	55.6456	112.7617	54.6639	112.6271
UIPCC	44.0289	113.5931	42.1972	108.7312	45.4457	106.5104	49.1318	108.1776	46.6078	108.8799
AMF	55.4219	116.2986	45.0937	109.2138	40.0813	**99.3292**	36.3137	**92.9208**	35.0542	92.7732
PMF	44.7542	111.3134	43.5078	107.6883	43.0480	107.6804	42.4471	106.6799	41.0839	104.5442
NRCF	54.8712	113.1567	53.8894	**105.0316**	52.4532	103.5014	52.0518	103.3833	52.1825	103.2707
NIMF	44.6099	110.9525	43.4722	108.5021	42.2466	105.8949	37.9826	97.4636	**33.4272**	**90.5786**
NeuMF	46.6071	120.1839	46.6044	120.1765	46.6050	120.1794	46.6069	120.1869	46.6085	120.1939
LDCF	43.1368	**109.5748**	42.7886	126.9160	41.4971	120.7278	40.1637	119.9376	38.5090	110.6470
BGCL	**42.7198**	110.7409	**40.2451**	105.2465	**37.0181**	100.9784	**35.3380**	95.8046	33.9775	90.6157
TEPP	**40.3329**	**109.4105**	**37.5284**	**103.4707**	**35.1827**	**98.1631**	**33.3114**	**92.7275**	**32.2278**	**88.5879**
Gains	5.59 %	0.15 %	6.75 %	1.49 %	4.96 %	1.17 %	5.73 %	0.21 %	3.59 %	2.20 %
P-UIPCC	58.7016	115.1683	56.5857	107.4806	55.4931	106.4743	53.2805	105.1682	52.5451	104.8075
P-PMF	55.6928	114.3494	53.3948	**106.4904**	52.2641	**103.0886**	51.1084	**102.9215**	50.3421	**102.4846**
Lap-UCF	58.1432	**113.5873**	55.6329	107.8394	54.9785	107.2318	54.5623	106.9171	54.2909	106.9086
FedNeuMF	**46.6075**	120.1872	**46.6046**	120.1793	**46.6039**	120.1801	**46.6050**	120.1858	**46.6078**	120.2011
TEPP-ϵ	**41.8204**	**113.1787**	**39.0398**	**107.1405**	**37.2299**	**102.2525**	**36.8553**	**101.8990**	**34.7639**	**95.1550**
Gains	10.27 %	0.36 %	16.23 %	−0.61 %	20.11 %	0.81 %	20.92 %	0.99 %	25.41 %	7.15 %

Table 6
Performance comparison of different QoS prediction methods on the D2-Sub-RT dataset.

Method	Density = 0.1 %		Density = 0.2 %		Density = 0.3 %		Density = 0.4 %		Density = 0.5 %	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.9641	2.0006	0.9144	1.8680	0.8987	1.8433	0.8911	1.8334	0.8886	1.8346
UPCC	0.9213	1.9125	0.9183	1.8923	0.8854	1.8159	0.8748	1.8407	0.8566	1.8258
UIPCC	0.8062	1.9607	0.7957	1.9178	0.7899	1.8864	0.7805	1.8486	0.7734	1.8188
AMF	0.8802	1.8729	0.7342	1.7443	0.6534	1.6780	0.6295	1.6932	0.6116	1.6959
PMF	0.7699	1.8770	0.7431	1.8378	0.7306	1.7849	0.7283	1.7785	0.7224	1.7698
NRCF	0.7304	1.6227	0.6959	1.5480	0.6848	1.5213	0.6787	1.5089	0.6606	1.4829
NIMF	0.7311	1.8602	0.7213	1.8495	0.7086	1.8212	0.7498	1.8987	0.7236	1.8613
NeuMF	0.6261	1.8752	0.6105	1.8712	0.6060	1.8670	0.6036	1.8439	0.5934	1.8405
LDCF	**0.5534**	**1.5143**	**0.5320**	**1.4703**	**0.5207**	**1.4894**	**0.5143**	**1.4664**	**0.5054**	**1.4664**
BGCL	0.5971	1.8690	0.5942	1.8645	0.5931	1.8476	0.5889	1.8428	0.5812	1.7615
TEPP	**0.5446**	**1.5091**	**0.4685**	**1.4318**	**0.4550**	**1.4026**	**0.4491**	**1.3959**	**0.4457**	**1.3875**
Gains	1.59 %	0.34 %	11.94 %	2.62 %	12.62 %	5.83 %	12.68 %	4.81 %	11.81 %	5.38 %
P-UIPCC	0.8759	1.9197	0.8327	1.8074	0.8032	1.7517	0.8087	1.7674	0.8068	1.7720
P-PMF	0.8615	1.9016	0.8097	1.7757	0.7745	1.7108	0.7754	1.7176	0.7692	1.7134
Lap-UCF	0.9065	**1.6729**	0.8997	**1.5551**	0.8801	**1.5239**	0.8791	**1.5161**	0.8696	**1.4928**
FedNeuMF	**0.6347**	1.9026	**0.6333**	1.9075	**0.6321**	1.9091	**0.6319**	1.9070	**0.6317**	1.9063
TEPP-ϵ	**0.5466**	**1.5110**	**0.4717**	**1.4434**	**0.4579**	**1.4146**	**0.4538**	**1.4141**	**0.4503**	**1.3912**
Gains	13.88 %	9.68 %	25.52 %	7.18 %	27.56 %	7.17 %	28.18 %	6.73 %	28.72 %	6.81 %

Table 7
Performance comparison of different QoS prediction methods on the D1 dataset.

Method	Density = 0.1 %		Density = 0.2 %		Density = 0.3 %		Density = 0.4 %		Density = 0.5 %	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.5667	1.2058	0.5626	1.1921	0.5623	1.1841	0.5602	1.1727	0.5591	1.1640
UPCC	0.5376	1.1407	0.5334	1.1289	0.5333	1.1221	0.5317	1.1126	0.5311	1.1058
UIPCC	0.4887	1.0885	0.4852	1.0771	0.4831	1.0677	0.4818	1.0600	0.4812	1.0530
AMF	0.3908	0.9063	0.3410	0.8504	0.3254	**0.8374**	0.3181	**0.8322**	0.3143	**0.8300**
PMF	**0.3434**	0.8864	0.3427	0.8835	0.3423	0.8808	0.3419	0.8788	0.3416	0.8770
NRCF	0.4077	**0.8525**	0.4026	**0.8426**	0.3982	0.8383	0.3979	0.8380	0.3973	0.8372
NIMF	0.3465	0.8949	0.3451	0.8921	0.3445	0.8905	0.3440	0.8893	0.3437	0.8887
NeuMF	0.3445	0.8906	**0.2973**	0.8565	**0.3010**	0.8506	**0.2985**	0.8538	**0.2956**	0.8500
LDCF	0.4340	1.2784	0.4305	1.2631	0.4320	1.2288	0.4311	1.1402	0.4257	1.0258
BGCL	0.4113	1.2239	0.4107	1.2231	0.4091	1.2100	0.4089	1.2015	0.4076	1.1950
TEPP	**0.3008**	**0.8399**	**0.2934**	**0.8378**	**0.2928**	**0.8222**	**0.2922**	**0.8215**	**0.2902**	**0.8176**
Gains	12.41 %	1.48 %	1.31 %	0.57 %	2.72 %	1.82 %	2.11 %	1.29 %	1.83 %	1.49 %
P-UIPCC	0.5317	1.1371	0.5309	1.1301	0.5257	1.1151	0.5213	1.1025	0.5198	1.0941
P-PMF	0.3767	0.9065	0.3657	0.8864	0.3616	0.8805	0.3593	0.8772	0.3584	0.8753
Lap-UCF	0.4503	**0.8793**	0.4362	**0.8586**	0.4307	**0.8523**	0.4292	**0.8513**	0.4283	**0.8497**
FedNeuMF	**0.3527**	0.9057	**0.3030**	0.8706	**0.3026**	0.8657	**0.2989**	0.8587	**0.2980**	0.8565
TEPP-ϵ	**0.3057**	**0.8407**	**0.2983**	**0.8392**	**0.2972**	**0.8353**	**0.2942**	**0.8266**	**0.2930**	**0.8231**
Gains	13.33 %	4.39 %	1.55 %	2.26 %	1.78 %	1.99 %	1.57 %	2.90 %	1.68 %	3.13 %

compared to collaborative filtering methods. AMF enhances online prediction capabilities by introducing an adaptive mechanism, but its performance degrades when the matrix density is 0.1 %, mainly affected by the extremely sparse data. NRCF uses an enhanced neighborhood model and a novel similarity calculation method to make progress in personalized QoS recommendations. NIMF combines traditional neighborhood methods with matrix factorization, and its performance is slightly improved compared to PMF. LDCF combines multi-layer perceptron (MLP) with a similarity adaptive correction mechanism to improve nonlinear modeling capabilities. NeuMF uses MLP and generalized matrix factorization to model the complex interactions between users and services, which to a certain extent makes up for the shortcomings of traditional methods. BGCL builds a dual-subgraph structure through graph contrastive learning, which strengthens the perception and modeling capabilities of user and service embedding features and improves prediction accuracy. The results in the table show that our solution improves MAE and RMSE by up to 8.70 % and 6.12 % on RT, and 6.75 % and 2.20 % on TP compared with the best performance of the comparison solution, respectively. TEPP achieves excellent prediction performance mainly due to two key designs: first, the introduction of a trust management mechanism and an enhanced similarity calculation method effectively quantify and mine trust relationships between users, thereby improving the reliability of neighborhood selection; second, an adaptive fusion model based on two weighting parameters is constructed to dynamically integrate user's past preferences and the trust information between users and services, thereby improving both prediction accuracy and generalization ability.

To compare and analyze the QoS prediction performance of the proposed methods in privacy-preserving scenarios, we set the privacy budget $\epsilon = 0.5$ in the comparative experiments and compared TEPP with four mainstream privacy-preserving schemes. The experimental results are presented in Tables 4 and 5, where TEPP demonstrates significant advantages in several evaluation metrics. Specifically, TEPP achieves at least 3.41 % and 3.98 % improvements in MAE and RMSE on RT, respectively. It also improves MAE by at least 10.27 % on TP. The improvement in the RMSE metric is relatively small, except for the case where TEPP performs worse than the baseline at 0.2 % matrix density, which still shows excellent performance. In summary, TEPP exhibits good QoS prediction performance even at this very low matrix density, and it is expected to maintain strong predictive capability as the matrix density increases. TEPP's excellent performance in privacy-preserving scenarios is primarily attributed to the adoption of the exponential mechanism, which ensures the protection of users' private information. Compared with other differential privacy methods (e.g., the Laplace mechanism), the exponential mechanism is more suitable for discrete selection tasks in service recommendation, as it protects user privacy while maintaining the accuracy of prediction results.

To further verify the generalization ability and scalability of the TEPP solution under different data scales, we conducted experiments on the D2-Sub-RT and D1 datasets, with the results presented in Tables 6 and 7. The results indicate that TEPP consistently achieves superior performance across different data scales.

As shown in Table 6, TEPP achieves improvements of 12.68 % and 5.83 % in MAE and RMSE, respectively, compared with the best performance of other competing methods on the D2-Sub-RT dataset. TEPP also demonstrates superior performance among privacy-preserving methods, achieving at least 13.88 % and 6.73 % improvements in MAE and in RMSE, respectively.

In addition, to perform the BGCL and LDCF comparison scheme on the D1 dataset, we obtain the user's country and Autonomous System Number (ASN) from the Client IP by using the IP Location Finder[2] and Team Cymru[3] tools, respectively, and then convert the country into a numeric code by using the United Nations M.49 Area Numeric Coding (UN M.49)[4], while the ASN retains only its numeric part. "Unknown" and other invalid entries are uniformly represented as "−1". The limited availability of only the user's country and ASN information results in the overall low prediction accuracy of BGCL and LDCF on this dataset. Nevertheless, Table 7 illustrates that TEPP still outperforms other state-of-the-art schemes on the D1 dataset, achieving at least 1.31 % and 0.57 % improvements in the MAE and RMSE on RT, respectively. Under privacy-preserving scenarios, the MAE and RMSE are further improved by at least 1.55 % and 1.99 %, respectively, which further verifies the superiority and scalability of TEPP in diverse data environments.

6.5.2. Ablation experiments and analysis

We conduct ablation experiments on the D2 dataset to verify the advantages of the fusion model through three experimental: a) Impact of the similarity weight factor; b) The advantages of the fusion model; c) Impact of the privacy budget.

Impact of similarity weight factor. To validate the benefits of incorporating the similarity weight factor, we compare TEPP schemes with and without this factor. We refer to the TEPP scheme without the

[2] https://www.iplocation.net/
[3] http://www.team-cymru.com/
[4] https://en.wikipedia.org/wiki/UN_M49

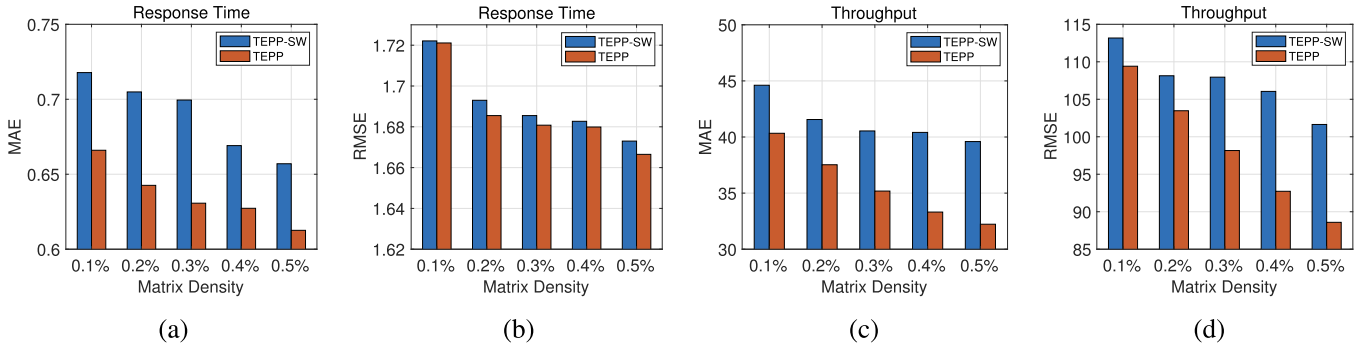

Fig. 3. Impact of similarity weight factor.

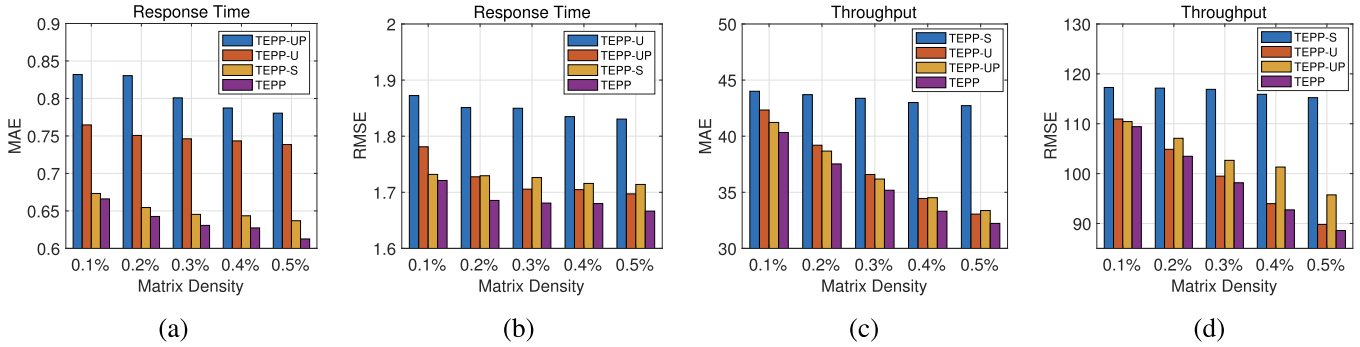

Fig. 4. The advantages of the fusion model.

similarity weight factor as TEPP-SW. We conducted experiments on both schemes for RT and TP, respectively, and presented the detailed results in Fig. 3. As shown in the figures, it is evident that all TEPP-SW values are higher than those of TEPP. In other words, the prediction performance of TEPP is superior to that of TEPP-SW. For instance, in Fig. 3(a), the MAE of TEPP-SW is 0.7178, compared to 0.666 for TEPP at a matrix density of 0.1 %. Similarly, in Fig. 3(c), the MAE of TEPP-SW is 44.6139, while that for TEPP is 40.3329 at a matrix density of 0.1 %. Thus, we conclude that introducing the similarity weight factor reduces the prediction error.

The advantages of the fusion model. To assess the individual contributions of user-based, user preference-based, and service-based QoS prediction components, we compare the proposed method with three of its variants. These variants are denoted as TEPP-U, TEPP-UP, and TEPP-S, where TEPP-U excludes the influence of users, TEPP-UP excludes the influence of user preferences, and TEPP-S excludes the influence of services. In TEPP-U ($\mu = 0$, $0 < \nu < 1$), the fusion model is expressed as $\mathcal{R}_{ij} = \nu \cdot R^s_{ij} + (1-\nu) \cdot R^p_{ij}$, which involves only the user preference-based and service-based prediction components. TEPP-UP excludes the user preference-based component by setting $\mu + \nu = 1$, resulting in the fusion model being transformed into $\mathcal{R}_{ij} = \mu \cdot R^u_{ij} + \nu \cdot R^s_{ij}$, which retains only the user-based and service-based prediction components. In TEPP-S ($0 < \mu < 1$, $\nu = 0$), the fusion model is represented as $\mathcal{R}_{ij} = \mu \cdot R^u_{ij} + (1-\mu) \cdot R^p_{ij}$, involving only the user-based and user preference-based components. We evaluated the prediction performance of these three variant schemes on RT and TP, and the specific experimental results are also presented in Fig. 4.

As illustrated in Fig. 4, TEPP achieves superior prediction performance over the three simplified variants. Specifically, compared with the best-performing variant TEPP-S, TEPP achieves at least 1.08 % and 0.62 % improvements in MAE and RMSE on RT, respectively. In terms of TP, compared with TEPP-UP, TEPP's MAE and RMSE are also improved by at least 2.17 % and 0.92 %, respectively. Overall, these results demonstrate that the fusion strategy proposed in this paper offers significant advantages in enhancing QoS prediction performance.

Impact of the privacy budget. To verify the impact of the privacy budget on the prediction accuracy of the TEPP scheme, we set the privacy budget ϵ to 0.1, 0.5, 0.9, and 1.0, and the experimental results are

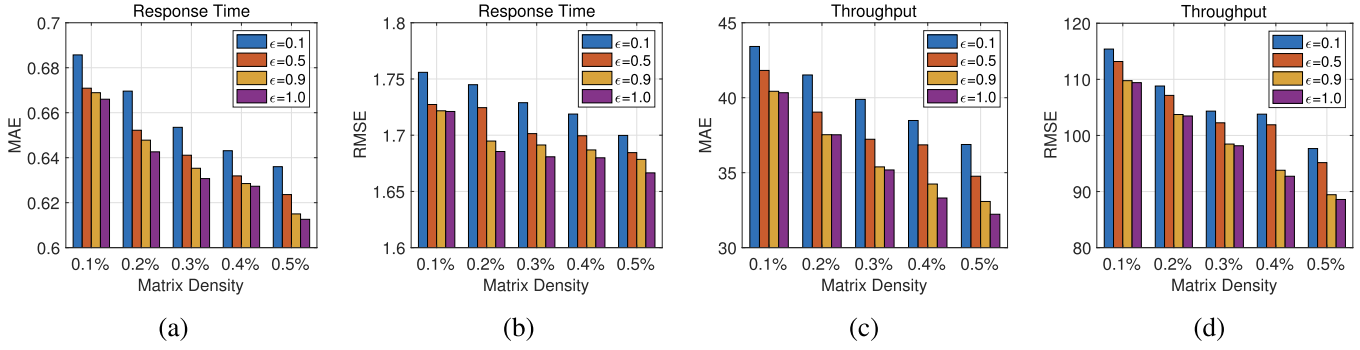

Fig. 5. Impact of the privacy budget.

shown in Fig. 5. A larger privacy budget ϵ results in less added noise and lower privacy protection, but it also leads to improved prediction accuracy. In other words, although a smaller ϵ can provide stronger privacy protection, it will lead to an increase in prediction error. As shown in Fig. 5, as ϵ increases, the MAE and RMSE values gradually decrease, indicating improved prediction accuracy. Taking response time as an example, the MAE and RMSE of $\epsilon = 0.1$ increased by at least 2.87 % and 1.96 % compared to $\epsilon = 1.0$, respectively; and when ϵ increased from 0.9 to 1.0, the MAE and RMSE decreased by no less than 0.19 % and 0.03 %, respectively. In terms of throughput prediction, the MAE and RMSE of $\epsilon = 1.0$ decreased by at least 7.11 % and 5.19 % compared to $\epsilon = 0.1$. In addition, as ϵ increases, the gap between the error values gradually narrows, indicating that the prediction results gradually become more accurate, further verifying that the TEPP scheme achieves a favorable trade-off between prediction performance and privacy.

6.6. Experimental analysis of bilateral trust model

We describe the bilateral trust model based on game theory in Section 5. Next, in this part, to assess the influence of various parameters on the trust game, we utilize MATLAB 2023b to conduct simulation experiments on the changing trends of the proposed trust game model under different parameters and the initial proportion of participants engaging in trusting behaviors.

6.6.1. Steady state strategy analysis

As depicted in Table 3, the bilateral trust model contains a total of eight steady state strategies. After analyzing these several situations, we conduct simulation experiments on the evolution trends of eight steady state strategies, as illustrated in Fig. 6. The specific parameter assignments are shown in Table 8. In Fig. 6(a), when $0 < \kappa < 1$, for strategies I, II-(1), and III-(1), the overall participation trust behavior will be shown. Even if the proportion of initial participants is extremely low, these participants will continue to enter the game of trusting behavior. This is because for these three strategies, the benefits obtained by players participating in the game are greater than the costs. For example: in strategy I, participants will participate in the trust game in a short period of time and quickly evolve into performing trust behaviors; in the two cases of strategy II-(1) and strategy III-(1), although participants will also participate in the trust game, the tendency to evolve into trusting behavior is slower than strategy I. This is because the benefits obtained by the players in strategy II-(1) and strategy III-(1) are smaller than those in strategy I. On the contrary, in strategy IV, although the proportion of initial players is quite high, it will not induce the interest of subsequent players to participate in the trust game, because the costs paid by the players under this strategy are more than the benefits obtained. In addition, we conclude that in the trust game, dynamic coefficients will also have an impact. In Fig. 6(b), when $\kappa > 1$, in strategy V, although the benefits obtained by the participants are less than those in strategy I, they will still evolve to perform trusting behavior, but the evolution trend speed is relatively The strategy is slower. For strategies VII-(1) and VIII-(2), as the cost of participants increases, participants will still enter the trust game with a very slow trend. This is because more rewards are given to these participants. They get more benefits. However, in strategy IX-(2), although the dynamic coefficient is larger than that in strategy IV, the cost paid is higher, which results in that no matter how large the initial proportion of participants is, they will eventually evolve into not participating in trusting behavior.

Table 8
Parameter settings for steady state strategies.

Strategies	(κ, B_i, A_i, C_i)	Strategies	(κ, B_i, A_i, C_i)
I	(0.5,5,1,1)	**II-(1)**	(0.6,3,1,2)
III-(1)	(0.8,3,2,4)	**VI**	(0.9,3,1,5)
V	(2,2,1,1)	**VII-(1)**	(2,2,6,7)
VIII-(2)	(2,3,4,8)	**IX-(2)**	(2,3,1,8)

6.6.2. Dynamic strategy analysis

In the analysis of dynamic strategies, we can see from Table 3 that there are five dynamic strategies. To validate the evolutionary process of these five strategies and analyze how participants' trust behaviors change with varying dynamic coefficients, we conducted simulations with different dynamic coefficients and initial participant ratios. The specific results are illustrated in Figs. 7 and 8. For $0 < \kappa < 1$, we analyze strategies II-(2) and III-(2), and present the specific results in Fig. 7. As illustrated in the figure, regardless of changes in the dynamic coefficient and p value, the final evolutionary trend does not exhibit a tendency toward engaging in trusting behavior. Nevertheless, it is worth noting that with the increase in both the k value and p value, the proportion of initial participants gradually increases but stabilizes quickly. Hence, under these two strategies, maximizing the dynamic coefficient value is

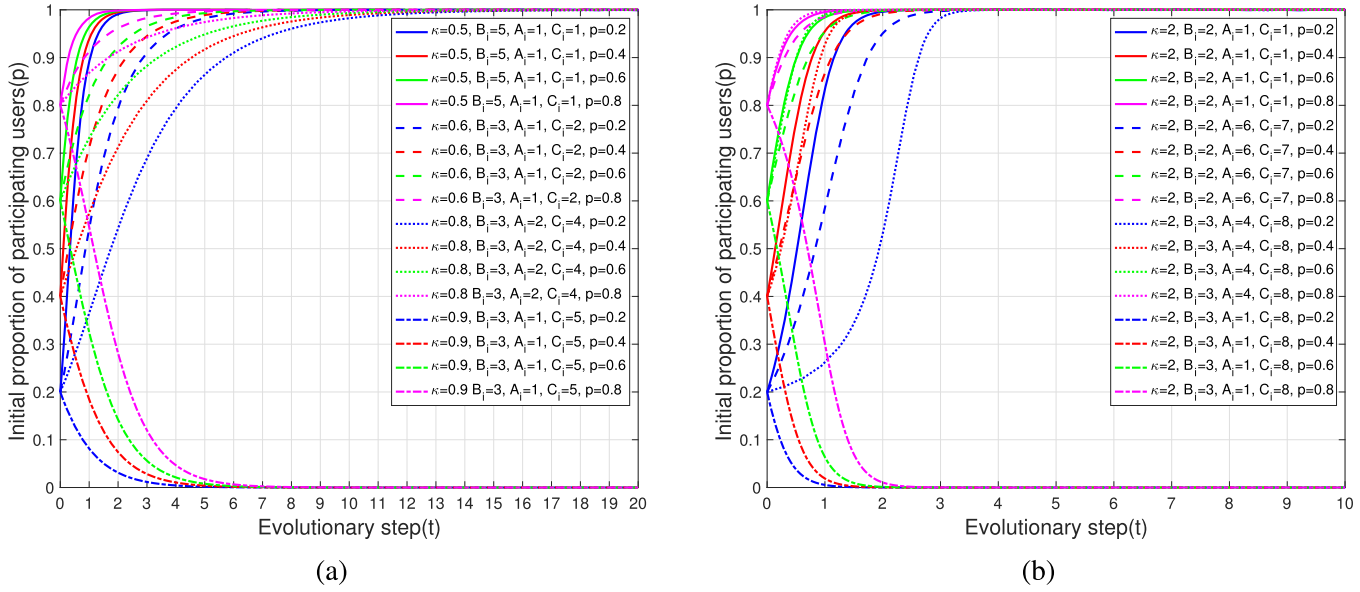

Fig. 6. Evolutionary graph of steady state. (a) $0 < \kappa < 1$; (b) $\kappa > 1$.

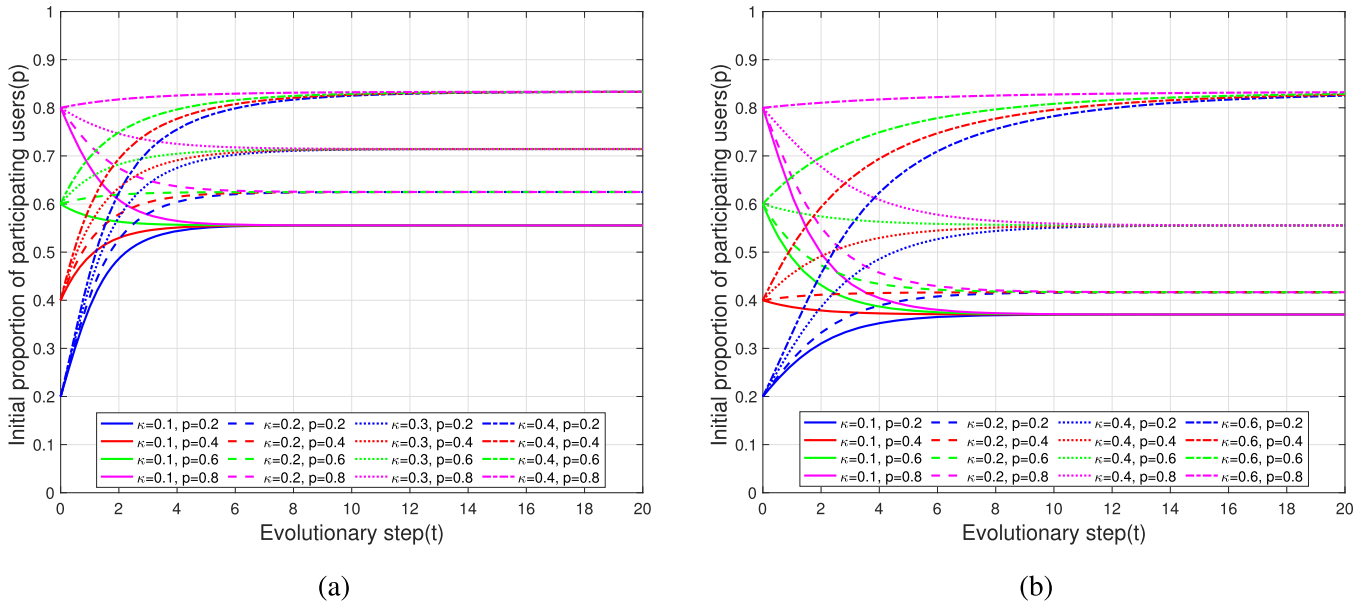

Fig. 7. Evolutionary graph of dynamic strategy with $0 < \kappa < 1$. (a) Strategy II-(2); (b) Strategy III-(2).

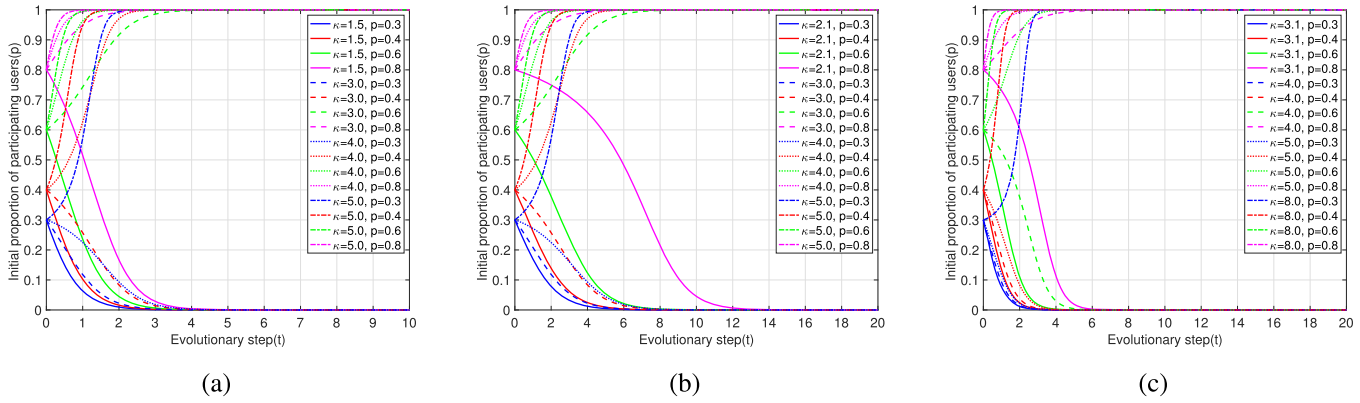

Fig. 8. Evolutionary graph of dynamic strategy with $\kappa > 1$. (a) Strategy VI; (b) Strategy VIII-(1); (c) Strategy IX-(1).

crucial to encourage more participants to engage in trusting behaviors. When $\kappa > 1$, the results of our dynamic strategy simulation are presented in Fig. 8, depicting three dynamic strategies: VI, VIII-(1), and IX-(1). Within these three strategies, participants with an initial participation rate of $p \in (0, p_3)$ evolve to refrain from trusting actions, while those with an initial participation rate of $p \in (p_3, 1)$ evolve to perform actions of trust.

7. Conclusion

This paper introduces a novel robust trust-enhanced privacy-preserving QoS prediction method, named TEPP, which synergistically integrates a Dirichlet-based reputation system, the exponential mechanism, and a bilateral trust model. TEPP effectively addresses the crucial trade-offs between user privacy, prediction performance, and system stability in Web service recommendation. Firstly, TEPP achieves excellent prediction accuracy by incorporating user trust values, actual preferences, and service trust values into an adaptive prediction model. Compared to the baseline scheme on common datasets, it achieves improvements of up to 8.70 % in MAE and 6.12 % in RMSE on RT, and improvements of up to 6.72 % and 2.20 % in MAE and RMSE on TP, respectively. Secondly, the incorporation of an exponential mechanism for privacy protection has been shown to effectively safeguard user information with minimal impact on prediction performance, and the prediction accuracy has increased by up to 3.73 % in MAE and 11.07 % in RMSE compared with other privacy protection schemes. Furthermore, the bilateral trust model embedded in TEPP has been shown to enhance system stability and encourage trust participation from both users and service providers when the initial proportion of participants performing trusted behaviors reaches a certain range. These results highlight the significant value of TEPP, which not only improves QoS prediction accuracy and effectively protects user privacy, but also promotes the construction of more reliable and user-centric service recommendation systems. Future research will explore the integration of blockchain technology to further enhance the credibility and system robustness of privacy-preserving QoS prediction in distributed service recommendation.

CRediT authorship contribution statement

Wei-wei Wang: Writing - original draft, Methodology, Software, Data curation, Conceptualization; **Wenping Ma:** Supervision, Funding acquisition; **Kun Yan:** Validation, Software.

Data availability

The data used in this work are publicly available.

Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We sincerely appreciate the anonymous reviewers and the editor for their constructive feedback and valuable suggestions. To improve the readability of the manuscript, we utilized Grammarly during the manuscript preparation process for grammar refinement and typo correction. This tool was solely used to enhance linguistic clarity without affecting the scientific content. This work was supported by the Open Fund Project of the State Key Laboratory of Intelligent Vehicle Safety Technology, China, under Grant No. IVSTSKL-202441, and the Key Industry Innovation Chain Project of Shaanxi Provincial Science and Technology Department, China, under Grant No. 2022ZDLGY03-08.

References

Badsha, S., Yi, X., Khalil, I., Liu, D., Nepal, S., Bertino, E., & Lam, K. Y. (2018a). Privacy preserving location-aware personalized web service recommendations. IEEE Transactions on Services Computing, 14(3), 791–804.

Badsha, S., Yi, X., Khalil, I., Liu, D., Nepal, S., & Lam, K.-Y. (2018b). Privacy preserving user based web service recommendations. IEEE Access, 6, 56647–56657.

Chang, Z., Ding, D., & Xia, Y. (2021). A graph-based QoS prediction approach for web service recommendation. Applied Intelligence, 51, 6728–6742.

Chen, J., Mao, C., & Song, W. W. (2023). QoS prediction for web services in cloud environments based on swarm intelligence search. Knowledge-Based Systems, 259, 110081.

Chen, Z., Bao, T., Qi, W., You, D., Liu, L., & Shen, L. (2024). Poisoning QoS-aware cloud API recommender system with generative adversarial network attack. Expert Systems with Applications, 238, 121630.

Chen, Z., Sun, Y., You, D., Li, F., & Shen, L. (2020). An accurate and efficient web service QoS prediction model with wide-range awareness. Future Generation Computer Systems, 109, 275–292.

Dwork, C., Roth, A. et al. (2014). The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4), 211–407.

Fiedler, M., Hossfeld, T., & Tran-Gia, P. (2010). A generic quantitative relationship between quality of experience and quality of service. IEEE Network, 24(2), 36–41.

Fkih, F. (2022). Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. Journal of King Saud University-Computer and Information Sciences, 34(9), 7645–7669.

Ghafouri, S. H., Hashemi, S. M., & Hung, P. C. K. (2020). A survey on web service QoS prediction methods. IEEE Transactions on Services Computing, 15(4), 2439–2454.

Hassan, H., El-Desouky, A. I., Ibrahim, A., El-Kenawy, E.-S. M., & Arnous, R. (2020). Enhanced QoS-based model for trust assessment in cloud computing environment. IEEE Access, 8, 43752–43763.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173–182).

Li, J., & Lin, J. (2020). A probability distribution detection based hybrid ensemble QoS prediction approach. Information Sciences, 519, 289–305.

Li, J., Wu, H., Chen, J., He, Q., & Hsu, C.-H. (2021). Topology-aware neural model for highly accurate QoS prediction. IEEE Transactions on Parallel and Distributed Systems, 33(7), 1538–1552.

Li, L., Li, S., & Zhao, S. (2014). QoS-aware scheduling of services-oriented internet of things. IEEE Transactions on Industrial Informatics, 10(2), 1497–1505.

Li, N., Lyu, M., Su, D., & Yang, W. (2016). Differential privacy: From theory to practice. Synthesis Lectures on Information Security, Privacy, & Trust, 8(4), 1–138.

Liu, A., Shen, X., Li, Z., Liu, G., Xu, J., Zhao, L., Zheng, K., & Shang, S. (2019). Differential private collaborative web services QoS prediction. World Wide Web, 22, 2697–2720.

Liu, A., Shen, X., Xie, H., Li, Z., Liu, G., Xu, J., Zhao, L., & Wang, F. L. (2020). Privacy-preserving shared collaborative web services QoS prediction. Journal of Intelligent Information Systems, 54, 205–224.

Liu, J., & Chen, Y. (2019). A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing. Knowledge-Based Systems, 174, 43–56.

Mezni, H. (2023). Web service adaptation: A decade's overview. Computer Science Review, 48, 100535.

Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. In Advances in neural information processing systems (pp. 1257–1264).

Perifanis, V., & Efraimidis, P. S. (2022). Federated neural collaborative filtering. Knowledge-Based Systems, 242, 108441.

Phelps, S., & Wooldridge, M. (2013). Game theory and evolution. IEEE Intelligent Systems, 28(04), 76–81.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on world wide web (pp. 285–295).

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., & Mei, H. (2007). Personalized QoS prediction forweb services via collaborative filtering. In IEEE International conference on web services (ICWS 2007) (pp. 439–446). IEEE.

Su, K., Xiao, B., Liu, B., Zhang, H., & Zhang, Z. (2017). TAP: A personalized trust-aware QoS prediction approach for web service recommendation. Knowledge-Based Systems, 115, 55–65.

Sun, H., Zheng, Z., Chen, J., & Lyu, M. R. (2012). Personalized web service recommendation via normal recovery collaborative filtering. IEEE Transactions on Services Computing, 6(4), 573–579.

Tao, Q., Chang, H.-y., Gu, C.-q., & Yi, Y. (2012). A novel prediction approach for trustworthy QoS of web services. Expert Systems with Applications, 39(3), 3676–3681.

Wang, W., Ma, W., & Yan, K. (2025). FSPPCFs: A privacy-preserving collaborative filtering recommendation scheme based on fuzzy C-means and shapley value. Complex & Intelligent Systems, 11(1), 107.

Wang, X., He, P., Zhang, J., & Wang, Z. (2020). QoS prediction of web services based on reputation-aware network embedding. IEEE Access, 8, 161498–161508.

Xie, Y., Guo, Y., Mi, Z., Yang, Y., & Obaidat, M. S. (2019). Loosely coupled cloud robotic framework for QoS-driven resource allocation-based web service composition. IEEE Systems Journal, 14(1), 1245–1256.

Xu, C., Wang, J., Zhu, L., Zhang, C., & Sharif, K. (2019). PPMR: A privacy-preserving online medical service recommendation scheme in eHealthcare system. IEEE Internet of Things Journal, 6(3), 5665–5673.

Xu, Y., Yin, J., Deng, S., Xiong, N. N., & Huang, J. (2016). Context-aware QoS prediction for web service recommendation and selection. Expert Systems with Applications, 53, 75–86.

Zhang, P., Huang, W., Chen, Y., Zhou, M., & Al-Turki, Y. (2024a). A novel deep-learning-based QoS prediction model for service recommendation utilizing multi-stage multi-scale feature fusion with individual evaluations. IEEE Transactions on Automation Science and Engineering, 21(2), 1740–1753.

Zhang, P., Jin, H., Dong, H., Song, W., & Bouguettaya, A. (2020). Privacy-preserving qos forecasting in mobile edge environments. IEEE Transactions on Services Computing, 15(2), 1103–1117.

Zhang, P., Ren, J., Huang, W., Chen, Y., Zhao, Q., & Zhu, H. (2024b). A deep-learning model for service QoS prediction based on feature mapping and inference. IEEE Transactions on Services Computing, 17(4), 1311–1325.

Zhang, Y., Yin, C., Wu, Q., He, Q., & Zhu, H. (2019). Location-aware deep collaborative filtering for service recommendation. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(6), 3796–3807.

Zheng, Z., & Lyu, M. R. (2010). Collaborative reliability prediction of service-oriented systems. In Proceedings of the 32nd ACM/IEEE international conference on software engineering-volume 1 (pp. 35–44).

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2011). QoS-aware web service recommendation by collaborative filtering. IEEE Transactions on Services Computing, 4(2), 140–152.

Zheng, Z., Li, X., Tang, M., Xie, F., & Lyu, M. R. (2020). Web service QoS prediction via collaborative filtering: A survey. IEEE Transactions on Services Computing, 15(4), 2455–2472.

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2012a). Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Transactions on Services Computing, 6(3), 289–299.

Zheng, Z., Zhang, Y., & Lyu, M. R. (2012b). Investigating QoS of real-world web services. IEEE Transactions on Services Computing, 7(1), 32–39.

Zhu, J., He, P., Zheng, Z., & Lyu, M. R. (2015). A privacy-preserving QoS prediction framework for web service recommendation. In 2015 IEEE International conference on web services (pp. 241–248). IEEE.

Zhu, J., He, P., Zheng, Z., & Lyu, M. R. (2017). Online QoS prediction for runtime service adaptation via adaptive matrix factorization. IEEE Transactions on Parallel and Distributed Systems, 28(10), 2911–2924.

Zhu, J., Li, B., Wang, J., Li, D., Liu, Y., & Zhang, Z. (2023). BGCL: Bi-subgraph network based on graph contrastive learning for cold-start QoS prediction. Knowledge-Based Systems, 263, 110296.

Zou, G., Li, T., Jiang, M., Hu, S., Cao, C., Zhang, B., Gan, Y., & Chen, Y. (2022). DeepTSQP: Temporal-aware service QoS prediction via deep neural network and feature integration. Knowledge-Based Systems, 241, 108062.

Zou, G., Wu, S., Hu, S., Cao, C., Gan, Y., Zhang, B., & Chen, Y. (2023). NCRL: Neighborhood-based collaborative residual learning for adaptive QoS prediction. IEEE Transactions on Services Computing, 16(3), 2030–2043.